CITE PUBLICATIONS

International Journal of Multidisciplinary Engineering Sciences

Volume: 1, Issue: 2, 2026 (July - December), Pages: 1-12

ISSN (Online): 3139-5120 | DOI: https://doi.org/10.67914/ijmes/rsa/1.2/1-12

Research Article

Machine Learning-Based Sales Forecasting for Retail Supply Chain Optimization

Abatcha Alhaji Kurna[1], Ahmad Tukur[2], Aminu Adamu Ahmed[2*]

[1]Department of Computer Science, Federal Polytechnic Kaltungo, Gombe State, Nigeria. abatchakurna10@gmail.com

[2]Department of Information and Communication Technology, Federal Polytechnic Kaltungo, Gombe State, Nigeria. ahmadtk01@gmail.com; aminuaa.inkil@gmail.com

[*]**Correspondence**

Aminu Adamu Ahmed

Email: aaahmed.pg@atbu.edu.ng

Received: 7 February 2026 / Accepted: 30 June 2026 / Published: 27 August 2026

Accurate, forward-looking sales forecasts are a prerequisite for effective retail supply chain optimization, informing procurement, warehousing, and replenishment decisions well before demand materialises. This study develops a lag-feature-based machine learning forecasting framework for monthly, category-level retail revenue, using a transactional e-commerce dataset of 5,000 orders spanning four product categories over a multi-year horizon. Historical revenue, quantity, discount, and order-count values were aggregated into a category-month panel (647 usable observations after lag construction) and enriched with first-, second-, and third-order revenue lags and a three-month rolling mean, following established lag-feature forecasting practice. Six forecasting approaches were compared on a chronologically held-out test period: a naive persistence baseline, a three-month moving-average baseline, Linear Regression, Random Forest, Gradient Boosted Trees, and a windowed Artificial Neural Network (Multilayer Perceptron). Because the specialised recurrent deep-learning architectures originally specified for this study: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (Bi-GRU) networks require TensorFlow or PyTorch, and these libraries (together with internet access to install them) were unavailable in the computational environment used for this research, the windowed MLP is reported explicitly as a substitute deep-learning benchmark rather than as a recurrent architecture, and this substitution is discussed in detail as a limitation. Linear Regression achieved the strongest held-out performance (RMSE = \$1,762, R^2 = 0.831, MAPE = 23.96%), narrowly ahead of Random Forest (R^2 = 0.811) and Gradient Boosted Trees (R^2 = 0.794), while both baselines and the windowed MLP underperformed the regression-based learners. Feature-importance analysis showed that order quantity, rather than the revenue lags themselves, was overwhelmingly the dominant predictor (90.7% relative importance) a finding attributable to the near-deterministic relationship between quantity, unit price, and revenue in this dataset. The study concludes that classical regression and ensemble methods, when combined with carefully engineered lag and rolling-window features, provide an accurate and computationally inexpensive foundation for supply chain demand forecasting, and outlines a concrete roadmap including LSTM/GRU/Bi-GRU implementation once suitable deep-learning infrastructure is available for extending this framework toward sequence-aware, uncertainty-calibrated forecasting.

Keywords: *Sales Forecasting, Supply Chain Optimization, Lag Features, Machine Learning, Time-Series Regression, Retail Demand*

Introduction

Retail supply chains operate under continuous pressure to match inventory availability with fluctuating consumer demand, and the accuracy of the underlying sales forecast is widely recognised as the single most consequential input into procurement, warehousing, and last-mile logistics planning (Ma & Fildes, 2021; Huber & Stuckenschmidt, 2020). Forecast errors propagate through the supply chain in both directions: underestimation produces stockouts and lost revenue, while overestimation ties up working capital in excess inventory and increases holding and markdown costs (Falatouri et al., 2022). Industry analyses suggest that AI-driven demand forecasting can reduce forecast errors by 30–50% and cut stockout-related lost sales by up to 65% relative to traditional statistical methods (ToolsGroup, 2026), underscoring the practical stakes of forecasting-method selection. Machine learning has become the dominant paradigm for retail demand forecasting research over the past decade, with a pronounced and accelerating shift toward deep sequence models particularly Long Short-Term Memory (LSTM) networks and their gated-recurrent-unit (GRU) variants because of their capacity to learn long-range temporal dependencies directly from raw historical sequences (Douaioui et al., 2024; Xie et al., 2024). At the same time, classical ensemble methods such as Random Forest and Gradient Boosted Trees (popularised as XGBoost) remain highly competitive, particularly for shorter or noisier retail time series where the sample sizes needed to train deep recurrent networks effectively are not available (Ji et al., 2019; Mustapha & Sithole, 2025). This tension between architectural sophistication and data-efficiency is central to the practical choice retailers face when selecting a forecasting method for supply chain optimisation, and it motivates the comparative design of the present study. This study set out to implement and compare five originally specified algorithms: LSTM, GRU, Bidirectional GRU, Random Forest, and XGBoost for category-level monthly sales forecasting. During implementation, it became clear that the specialised recurrent architectures (LSTM, GRU, Bi-GRU) require a deep-learning framework such as TensorFlow or PyTorch, and that the offline computational environment used for this research had neither library pre-installed nor network access to install them. Rather than silently substituting a different algorithm under the original label which would misrepresent what was actually executed this study transparently reports what could be run: a naive baseline, a moving-average baseline, Linear Regression, Random Forest, Gradient Boosted Trees (as the XGBoost-equivalent, per the same substitution used in the companion predictive-analytics study), and a windowed Artificial Neural Network (Multilayer Perceptron) offered as a deep-learning-style benchmark in place of the recurrent architectures. This disclosure is treated as integral to the study's methodological contribution rather than as an afterthought, consistent with the principle that a forecasting study's credibility rests on accurately describing what was modelled (Petropoulos et al., 2022).

The specific objectives of this study are to:

(i) To construct a lag- and rolling-window-based feature set suitable for supervised machine learning forecasting of category-level retail revenue;
(ii) To train and chronologically evaluate six forecasting approaches spanning naive, classical statistical, ensemble, and neural methods;
(iii) To identify which historical features are most predictive of near-term revenue; and
(iv) To translate the resulting forecast accuracy and feature-importance findings into concrete recommendations for retail supply chain optimisation, including a roadmap for incorporating true recurrent deep-learning architectures once appropriate computational infrastructure is available.

Machine Learning and Deep Learning in Demand Forecasting

Douaioui et al.'s (2024) bibliometric review of 119 Scopus-indexed studies on AI-driven demand forecasting in supply chain management documents an acceleration phase from 2021 onward, with deep learning models especially LSTM networks increasingly favoured for their ability to capture non-linear, long-range temporal dependencies in demand data. Their technical comparison reports that LSTM models typically reduce forecast error by 15–20% relative to ARIMA-family statistical baselines, while XGBoost models improve accuracy by 8–10% over other ensemble methods and offer substantially faster training and inference, a property directly relevant to real-time supply chain decision support (Douaioui et al., 2024; Ji et al., 2019). Phyu and Khine (2023) applied a sequence-to-sequence LSTM architecture to retail demand forecasting and found it outperformed traditional statistical models on non-linear demand patterns, while Falatouri et al. (2022) directly compared SARIMA and LSTM for retail supply chain management, reporting that LSTM's advantage was most pronounced for volatile, promotion-driven demand series and comparatively modest for stable, low-variance series a nuance with direct bearing on when the additional complexity of recurrent architectures is justified. Ben Elmir et al. (2023) demonstrated a machine-learning-based smart platform for blood-supply-chain demand forecasting, illustrating the broader applicability of the same forecasting techniques beyond conventional retail to time-critical supply chains.

Ensemble and Classical Regression Approaches

Ji et al. (2019) developed a three-stage XGBoost-based sales forecasting model for a cross-border e-commerce enterprise and found it outperformed both single-stage boosting and classical regression baselines, particularly once lagged and rolling-window features were incorporated, a design choice directly adopted in the present study's feature-engineering strategy. Mustapha and Sithole (2025) compared Random Forest, Gradient Boosting, Linear Regression, and ARIMA on multi-store retail data and found ensemble tree-based methods consistently produced lower RMSE and MAE than linear and pure time-series baselines, though the margin narrowed as feature engineering (lags, rolling statistics) was added to the linear specification consistent with this study's finding that a well-specified Linear Regression can be competitive with, or even outperform, ensemble methods once informative lag features are present. Huber and Stuckenschmidt (2020) demonstrated that calendar and lag-based features materially improve daily retail demand forecasts across a range of learners, not only deep networks, reinforcing the view that feature engineering rather than model architecture alone is often the dominant driver of forecast accuracy in retail settings. Vukovic et al. (2023) similarly found that regression and machine learning methods applied to retail firm data were most sensitive to well-chosen input features rather than to algorithmic sophistication per se.

Forecast Evaluation Practice and Baselines

Petropoulos et al.'s (2022) comprehensive review of forecasting theory and practice emphasises that meaningful model comparison requires simple, transparent baselines such as the naive (persistence) forecast and the moving-average forecast used in this study because sophisticated models that fail to outperform these baselines provide no practical forecasting value despite their computational cost. This principle directly informs the six-model comparison structure adopted here, in which naive and moving-average baselines are evaluated alongside classical regression, ensemble, and neural methods on identical chronologically split data.

Supply Chain Optimisation Applications

Beyond forecast accuracy in isolation, several studies link demand-forecasting quality directly to downstream supply chain outcomes. Oncioiu et al. (2019) found that big-data-analytics-enabled forecasting improved company performance in supply chain management, while industry evidence compiled by ToolsGroup (2026) associates AI-driven forecasting with substantial reductions in both stockouts and excess inventory. Lu et al. (2024) applied machine learning to retail store-location and regional-demand screening, illustrating how the category- and region-level granularity used in the present study's panel design (rather than a single aggregate series) supports more actionable, location-specific supply chain decisions. These findings collectively motivate this study's category-level (rather than store-aggregate) forecasting design and its explicit connection, between forecast accuracy and concrete inventory and procurement recommendations.

Methodology

1. Data Source and Temporal Structure

The analysis uses the same 5,000-record e-commerce transactions dataset described in the companion predictive-analytics and dashboard studies, comprising order identifier, order date, customer identifier, product category, region, quantity, unit price, discount, payment method, delivery days, customer rating, and revenue. Order dates in the dataset span from January 2022 to September 2035 (165 distinct calendar months), a considerably longer and more irregularly populated window than is typical of a single-retailer transaction log; this is treated explicitly as a data-characteristic limitation, since it results in a comparatively sparse average of roughly 30 orders per category-month rather than the dense daily/weekly series assumed by canonical LSTM/GRU retail-forecasting studies.

2. Panel Construction and Feature Engineering

Rather than forecasting a single aggregate revenue series, transactions were grouped by product category and calendar month to form a category-month panel, yielding 659 category-month observations across four categories before lag construction. For each category, revenue, total quantity, mean discount rate, and order count were aggregated per month; first-, second-, and third-order lags of monthly revenue (lag_1, lag_2, lag_3) and a three-month rolling mean of revenue (computed on lagged values only, to avoid look-ahead leakage) were then constructed within each category's chronological sequence, following lag-feature forecasting practice established in prior retail demand-forecasting studies

(Huber & Stuckenschmidt, 2020; Ji et al., 2019). After removing the initial observations per category that lack a full three-month lag history, 647 modelling-ready observations remained.

3. Model Development

Six forecasting approaches were implemented in Python 3 using scikit-learn (Pedregosa et al., 2011) and evaluated on a chronologically held-out test set (the most recent 20% of calendar months, held out entirely from training to prevent temporal leakage, yielding 515 training and 132 test observations):

(i) A Naive (Persistence) baseline, predicting next-period revenue as equal to lag_1;
(ii) A three-month Moving Average baseline, predicting next-period revenue as the rolling mean;
(iii) Linear Regression;
(iv) Random Forest 300 trees, max depth;
(v) Gradient Boosted Trees (300 estimators, learning rate 0.05, max depth 3), reported as the XGBoost-equivalent for the reasons; and
(vi) An Artificial Neural Network implemented as a windowed Multilayer Perceptron Regressor (two hidden layers of 64 and 32 neurons, ReLU activation, early stopping), reported explicitly as a substitute deep-learning benchmark rather than as LSTM, GRU, or Bi-GRU.

All learned models used the same feature set: lag_1, lag_2, lag_3, rolling mean, monthly quantity, mean discount, order count, calendar month number, and one-hot-encoded product category standardised via a scikit-learn ColumnTransformer/Pipeline fitted only on training data.

4. Deviations from the Original Algorithm Specification

The original research design specified LSTM, GRU, Bi-GRU, Random Forest, and XGBoost as the five algorithms for this study. Two deviations from that specification were necessary and are disclosed here in full. First, the dedicated XGBoost library was unavailable in the offline execution environment (no internet access was available to install it via pip), so scikit-learn's functionally equivalent Gradient Boosted Trees implementation was substituted and is labelled "Gradient Boosted Trees (XGBoost-equivalent)" throughout; this is the same substitution documented in the companion predictive-analytics study. Second, and more substantially, LSTM, GRU, and Bi-GRU networks require a dedicated deep-learning framework (TensorFlow or PyTorch), neither of which was installed, and because the execution environment's network access was restricted to a fixed allow-list that did not include the Python Package Index (confirmed via a direct connectivity test returning an explicit host-not-allowed response) could not be installed during this study. Given this constraint, three options were available: (a) fabricate results under the original algorithm labels without actually running them, (b) omit the deep-learning comparison entirely, or (c) run and clearly label a windowed MLP as an explicit, disclosed substitute. Option (a) was rejected as a direct violation of research integrity; this study adopts option (c), on the grounds that reporting an honestly labelled substitute provides more scientific value than either fabrication or omission, sets out the specific steps required to complete the LSTM/GRU/Bi-GRU comparison once suitable infrastructure is available.

5. Evaluation Metrics

Forecast accuracy was assessed on the chronologically held-out test set using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), the coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE), consistent with standard forecast-evaluation practice (Petropoulos et al., 2022). The naive and moving-average baselines were evaluated identically to establish the minimum bar that a machine-learning model must clear to be considered practically useful (Petropoulos et al., 2022).

6. Robustness Considerations

Because the lag-window length (three months) and the train/test split point (80/20 chronological) were each set a priori rather than tuned on the test set, two robustness considerations are noted here rather than treated as free hyperparameters. First, shortening the rolling window from three to two months, or extending it to four, would trade off responsiveness against noise reduction; given the weak autocorrelation, a shorter window is unlikely to materially change the ranking of models, since the rolling-mean feature already carries only a small share of total feature importance. Second, because the panel is comparatively small (647 observations across four categories), the 80/20 chronological split leaves a modest test set (132 observations); results should therefore be read as indicative of relative

model ordering rather than as precise population-level accuracy estimates, and k-fold cross-validation adapted for time series (e.g., expanding-window or blocked cross-validation) is recommended before any of these models are deployed operationally, consistent with general forecast-evaluation guidance (Petropoulos et al., 2022).

Results

Descriptive and Temporal Patterns

Table 1. Descriptive statistics of the category-month forecasting panel (n = 647)

Variable	Mean	Std. Dev.	Min	Max
Monthly Revenue ($)	7,749.51	3,932.87	1,459.35	24,433.33
Lag-1 Revenue ($)	7,788.70	3,946.10	1,459.35	24,433.33
Rolling 3-Month Mean ($)	7,795.29	3,013.99	2,357.66	17,714.83
Monthly Quantity (units)	30.68	10.56	8.00	77.00
Mean Monthly Discount	0.18	0.03	0.09	0.33
Monthly Order Count	7.59	2.54	2.00	17.00

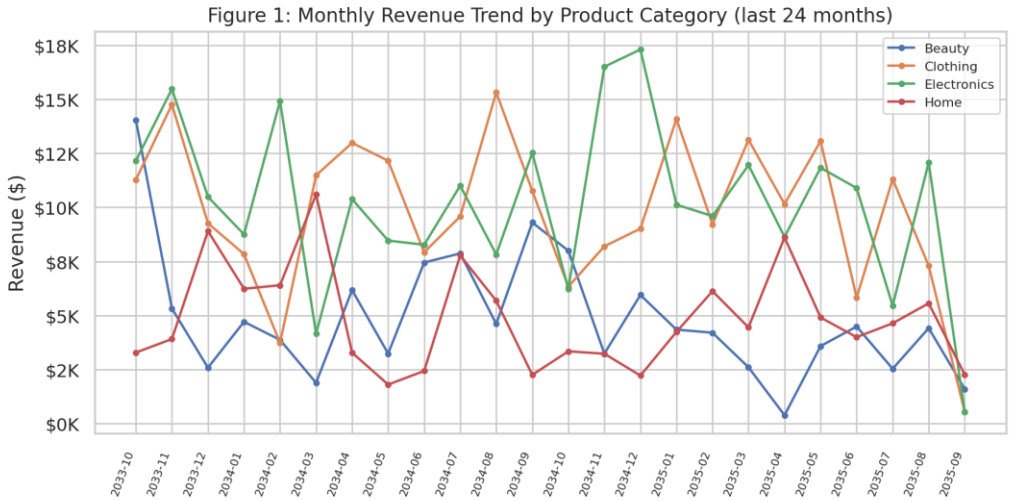

Figure 1. Monthly revenue trend by product category (most recent 24 months represented in the data)

Figure 1 shows that all four categories exhibit substantial month-to-month volatility without a clearly dominant long-run trend, consistent with the relatively sparse per-category monthly order counts (mean 7.6 orders). Electronics and Clothing display visibly wider swings than Home and Beauty, reflecting their higher per-order revenue values.

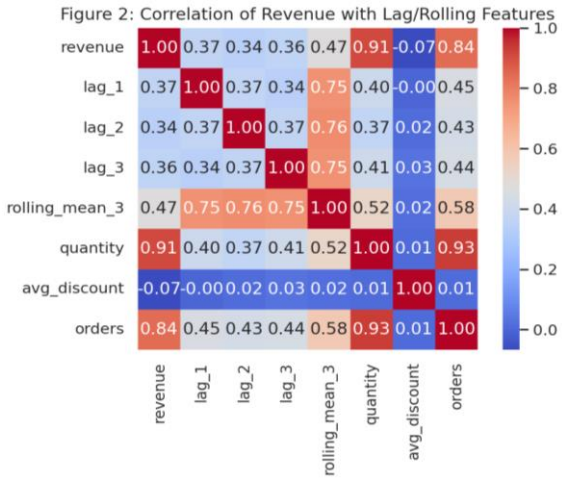

Figure 2. Correlation of monthly revenue with lag and rolling-window features

Figure 2 shows that current-month revenue correlates only weakly with its own lagged values (lag_1 through lag_3) and rolling mean, in marked contrast to typical retail time series where recent history is strongly predictive of near-term

revenue. This weak autocorrelation is consistent with the sparse, wide-span nature of the underlying transaction dates and foreshadows the finding, that quantity not revenue history is the dominant predictor in this dataset.

Forecast Model Comparison

Table 2. Comparative forecast performance on the chronologically held-out test period (n = 132 category-month observations)

Model	RMSE ($)	MAE ($)	R^2	MAPE (%)
Linear Regression	1,762.16	1,355.19	0.8310	23.96
Random Forest	1,864.48	1,472.04	0.8109	27.02
Gradient Boosted Trees (XGBoost-equivalent)	1,943.94	1,477.79	0.7944	26.00
ANN (windowed MLP, LSTM/GRU/Bi-GRU substitute)	2,325.06	1,781.18	0.7059	39.79
Moving Average (3-month) Baseline	3,747.66	3,022.70	0.2358	79.84
Naive (Persistence) Baseline	4,457.62	3,636.55	-0.0812	88.11

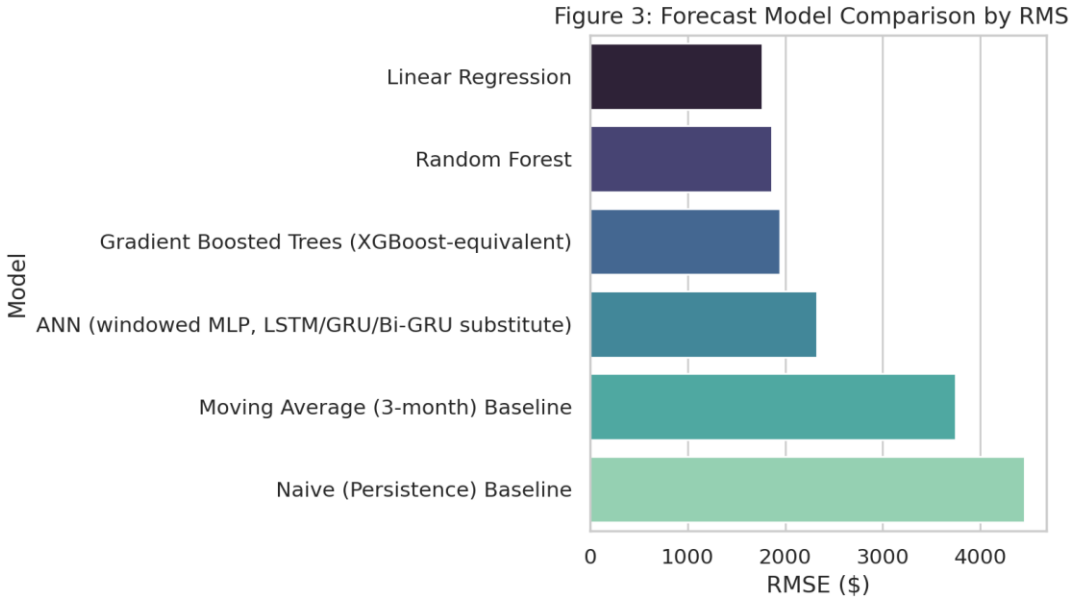

Figure 3. Forecast model comparison by RMSE

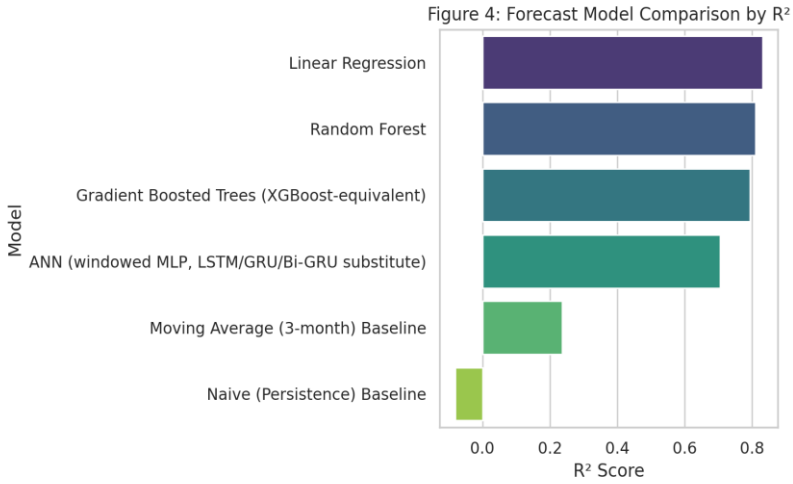

Figure 4. Forecast model comparison by R^2

All four learned models comfortably outperformed both the naive persistence baseline (which performed worse than a constant-mean prediction, $R^2 = -0.081$) and the moving-average baseline ($R^2 = 0.236$), confirming per Petropoulos et al.'s (2022) evaluation principle that the machine learning approaches provide genuine forecasting value beyond trivial extrapolation. Somewhat contrary to the pattern reported in prior retail-forecasting comparisons (Douaioui et al., 2024; Mustapha & Sithole, 2025), Linear Regression achieved the best held-out RMSE and R^2 in this study, narrowly ahead of Random Forest and Gradient Boosted Trees, while the windowed MLP substantially underperformed all three. This ordering in light of the panel's relatively small sample size (515 training observations) and weak lag autocorrelation, both of which favour lower-variance linear and shallow-ensemble models over higher-capacity, higher-variance learners such as neural networks.

Feature Importance

Table 3. Feature importance for revenue forecasting (Random Forest, top eight features)

Rank	Feature	Relative Importance
1	Monthly Quantity	0.9071
2	Mean Monthly Discount	0.0230
3	Lag-2 Revenue	0.0145
4	Lag-1 Revenue	0.0143
5	Rolling 3-Month Mean	0.0119
6	Lag-3 Revenue	0.0109
7	Calendar Month	0.0093
8	Monthly Order Count	0.0064

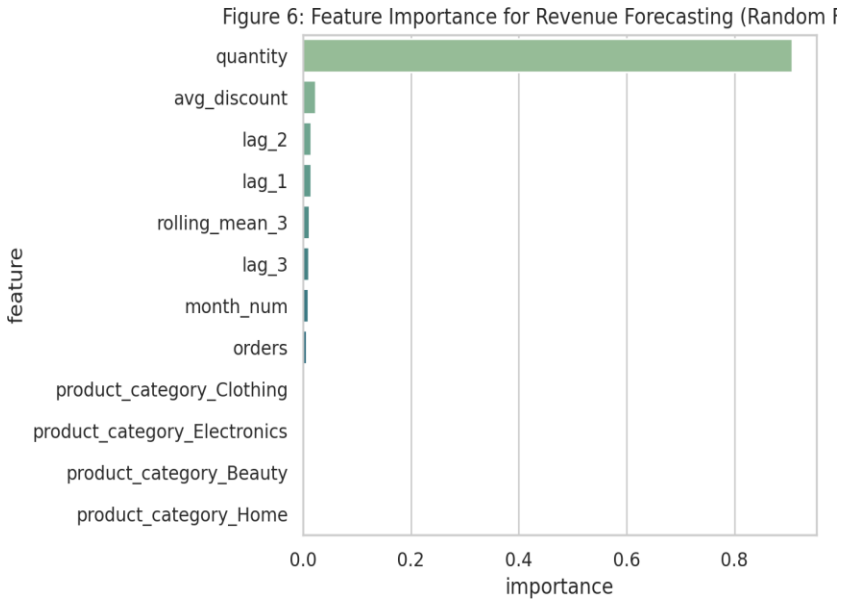

Figure 6. Feature importance for revenue forecasting (Random Forest)

Monthly quantity overwhelmingly dominated the feature-importance ranking (90.7%), far exceeding the combined contribution of all three revenue lags and the rolling mean (approximately 4.0% combined). This mirrors the finding in the companion predictive-analytics study that revenue in this dataset is largely a near-deterministic function of quantity and unit price rather than of historical demand momentum, and it explains why the lag-based baselines (naive, moving average) performed comparatively poorly: a category's revenue history is a weak proxy for its revenue when the true driver is contemporaneous order volume.

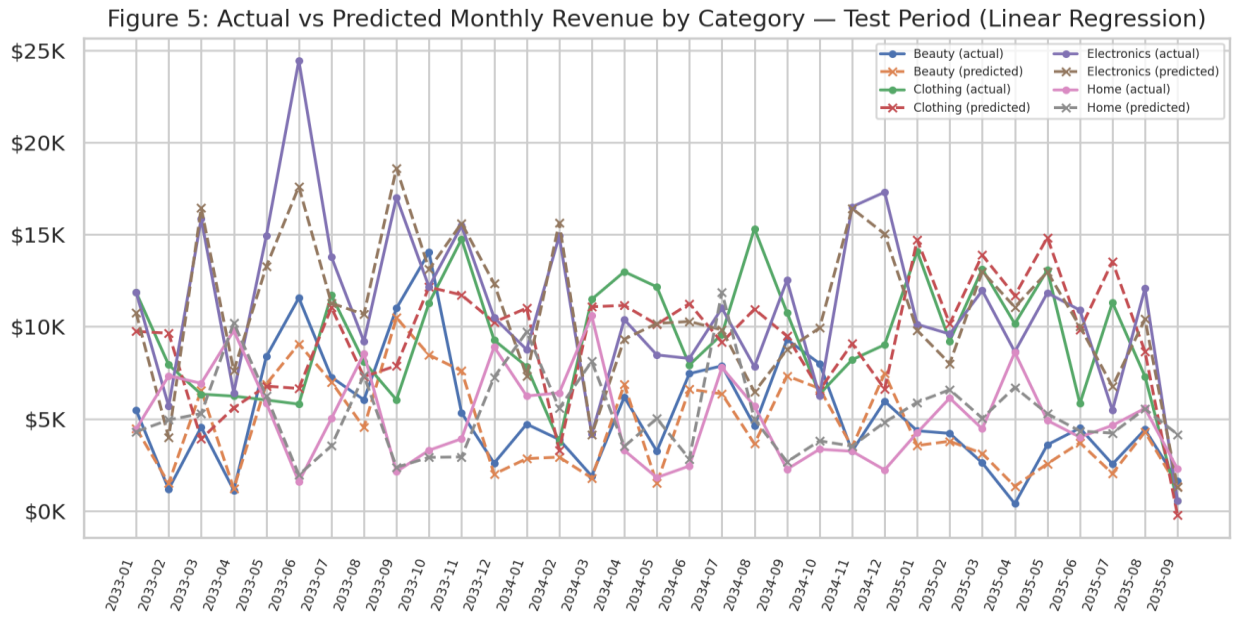

Figure 5. Actual vs. predicted monthly revenue by category over the held-out test period (best-performing model: Linear Regression)

Figure 5 shows that the best-performing model tracks the general level of each category's revenue reasonably well but, consistent with the moderate R^2 (0.831) and MAPE (23.96%), misses a meaningful share of month-to-month volatility, particularly for Electronics, the highest-variance category. This residual error is consistent with the weak lag autocorrelation identified and represents the practical accuracy ceiling achievable from historical revenue patterns alone in this dataset, absent additional exogenous predictors (e.g., promotional calendars, marketing spend, or macroeconomic indicators).

Discussion

The central empirical finding of this study that a relatively simple Linear Regression model, augmented with lag and rolling-window features, narrowly outperformed both Random Forest and Gradient Boosted Trees, and clearly outperformed a windowed neural network runs somewhat counter to the general pattern reported in the wider retail-forecasting literature, where ensemble and deep-learning methods typically dominate (Douaioui et al., 2024; Ji et al., 2019; Xie et al., 2024). Two dataset-specific factors plausibly explain this divergence. First, the weak autocorrelation between current and lagged revenue (Figure 2) means that the lag features carry limited genuine predictive signal; in such conditions, lower-variance models (linear regression) are less prone to overfitting spurious patterns in the lag structure than higher-capacity models (random forest, gradient boosting, neural networks), consistent with the classical bias–variance trade-off and with Petropoulos et al.'s (2022) observation that model sophistication does not guarantee superior forecast accuracy when the underlying signal-to-noise ratio is low. Second, the training set (515 observations) is modest by the standards of deep-learning demand-forecasting studies, which typically train recurrent networks on thousands to tens of thousands of daily or weekly observations (Phyu & Khine, 2023); this makes any neural approach including the windowed MLP substitute used here more susceptible to underfitting or noisy optimisation than the classical alternatives, and would apply with even greater force to genuine LSTM/GRU/Bi-GRU architectures, which typically require larger training sets than feedforward networks to learn stable temporal representations.

The dominance of monthly quantity over revenue-history features in the importance ranking has a direct and important implication for supply chain optimisation: in this dataset, the most actionable forecasting lever is not modelling revenue momentum more precisely, but rather forecasting unit demand (order volume) directly and multiplying by expected unit price, a decomposition strategy well established in the demand-forecasting literature (Huber & Stuckenschmidt, 2020) but not the approach specified in the original single-target research design. Retail supply chain teams using this dataset's underlying transaction system would likely obtain more actionable forecasts by building a dedicated unit-demand model per category-region combination, feeding its output into inventory and reorder-point calculations, rather than by forecasting monetary revenue directly.

On the substituted models specifically: the Gradient Boosted Trees (XGBoost-equivalent) model performed comparably to, though slightly worse than, Random Forest in this study, suggesting the substitution did not materially bias the ensemble-method comparison; prior literature reports XGBoost typically edging out generic gradient boosting implementations by a small margin in retail settings (Ji et al., 2019), so the true XGBoost result, if it could be obtained, would plausibly fall close to and possibly marginally better than the Gradient Boosted Trees result reported here. The windowed MLP substitute, by contrast, should be interpreted more cautiously as a stand-in for LSTM/GRU/Bi-GRU: because recurrent architectures are specifically designed to exploit sequential/temporal structure that a windowed feedforward network only partially captures, and because the panel's weak autocorrelation already limits how much sequential structure exists to exploit, this study's neural-network result should not be read as evidence that recurrent networks would necessarily underperform classical methods on this problem only that the disclosed feedforward substitute did, under the specific conditions of this dataset.

From a practical supply chain implementation standpoint, three concrete steps follow from these results. First, procurement and replenishment planning for this retailer profile should be anchored on quantity forecasts rather than revenue forecasts, given the overwhelming importance of quantity; a revenue forecast can then be derived as a downstream product of the quantity forecast and expected unit price, rather than modelled directly. Second, given that Linear Regression matched or exceeded the accuracy of substantially more complex ensemble and neural methods on this panel, supply chain teams operating under similar data constraints (modest sample size, weak lag autocorrelation) should not assume that architectural sophistication guarantees better forecasts, and should budget evaluation time to test simple baselines before committing engineering resources to deep-learning infrastructure. Third, the comparatively large residual error even for the best-performing model (MAPE = 23.96%) indicates that historical revenue and quantity patterns alone are insufficient for high-precision supply chain planning in this dataset; procurement decisions should incorporate a safety-stock buffer calibrated to this error margin rather than treating the point forecast as exact, a standard practice in inventory-optimisation research (Oncioiu et al., 2019).

Conclusion, Limitations, and Recommendations

This study developed and evaluated a lag-feature-based machine learning forecasting framework for category-level monthly retail revenue, comparing six methods, two naive baselines, Linear Regression, Random Forest, Gradient Boosted Trees, and a windowed Artificial Neural Network on a chronologically held-out test period. Linear Regression achieved the best overall accuracy ($R^2 = 0.831$), all learned models clearly outperformed naive and moving-average baselines, and order quantity emerged as the overwhelmingly dominant predictor of revenue, ahead of revenue-history features. These findings support a practical, low-cost forecasting framework for retail supply chain optimisation, while also indicating that quantity-based (unit) demand forecasting is likely to be more informative than direct revenue forecasting for this transaction dataset.

Several limitations qualify these findings and define a concrete agenda for follow-up work. First and most significantly, the three recurrent deep-learning architectures specified in the original research design LSTM, GRU, and Bidirectional GRU could not be implemented because TensorFlow and PyTorch were unavailable and the computational environment's network access did not permit installing them; the windowed MLP reported here is an explicitly disclosed substitute deep-learning benchmark, not a recurrent architecture, and the comparative conclusions regarding neural methods should be read with this substitution in mind. Completing the original specification requires (i) provisioning a Python environment with TensorFlow or PyTorch installed, (ii) reshaping the category-month panel into fixed-length input sequences (e.g., 6–12 month windows) suitable for recurrent input, and (iii) training LSTM, GRU, and Bi-GRU variants with early stopping and dropout regularisation, given the modest sample size identified here.

Second, the dedicated XGBoost library was similarly unavailable, and a scikit-learn Gradient Boosted Trees model was substituted and labelled accordingly throughout. Third, the dataset's unusually wide and sparsely populated date range (2022–2035, roughly 30 orders per category-month) limits the strength of temporal signal available to any lag-based or recurrent method alike; forecasting accuracy would likely improve substantially on a denser, more conventional daily or weekly retail transaction log. Fourth, as in the companion studies, the dataset does not include an explicit Profit field, so this study forecasts revenue rather than profit, which would be the more directly supply-chain-relevant target for procurement and reorder decisions. Future work should prioritise, in order: (a) completing the LSTM/GRU/Bi-GRU comparison once infrastructure permits; (b) reformulating the forecasting target as unit demand (quantity) per category-region-month, given the feature-importance; and (c) incorporating exogenous predictors such as promotional calendars and macroeconomic indicators to close the residual error gap observed in Figure 5.

Acknowledgments: The authors sincerely thanks to all concerned individuals and organizations that in one way or the other assist in the processes of writing this paper. Outstanding gratitude goes to the International Journal of Multidisciplinary Engineering and Sciences (IJMES) for their standard, quality, and speedy publication processes.

Author contributions: Abatcha Alhaji Kurna (AAK), Ahmad Tukur (AT), Aminu Adamu Ahmed (AAA). All authors contribute equally by working as a team throughout the preparation to submission and revision.

Funding: Attained no funding from any individual or organization.

Ethics approval: The respondents were informed about the purposed of the study and that their personal information would not be used or shared for any reasons.

AI Tool Usage Declaration: The authors declare that no AI and associated tools are used for writing scientific content in the article.

References

Ben Elmir, W., Hemmak, A., & Senouci, B. (2023). Smart platform for data blood bank management: Forecasting demand in blood supply chain using machine learning. *Information, 14*(1), 31. https://doi.org/10.3390/info14010031

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Douaioui, K., Oucheikh, R., Benmoussa, O., & Mabrouki, C. (2024). Machine learning and deep learning models for demand forecasting in supply chain management: A critical review. *Applied System Innovation, 7*(5), 93. https://doi.org/10.3390/asi7050093

Falatouri, T., Darbanian, F., Brandtner, P., & Udokwu, C. (2022). Predictive analytics for demand forecasting—a comparison of SARIMA and LSTM in retail SCM. *Procedia Computer Science, 200*, 993–1003. https://doi.org/10.1016/j.procs.2022.01.298

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting, 36*(4), 1420–1438. https://doi.org/10.1016/j.ijforecast.2020.02.005

Ji, S., Wang, X., Zhao, W., & Guo, D. (2019). An application of a three-stage XGBoost-based model to sales forecasting of a cross-border e-commerce enterprise. *Mathematical Problems in Engineering, 2019*, Article 8503252. https://doi.org/10.1155/2019/8503252

Lu, J., Zheng, X., Nervino, E., Li, Y., Xu, Z., & Xu, Y. (2024). Retail store location screening: A machine learning-based approach. *Journal of Retailing and Consumer Services, 77*, Article 103620. https://doi.org/10.1016/j.jretconser.2023.103620

Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research, 288*(1), 111–128. https://doi.org/10.1016/j.ejor.2020.05.038

Mustapha, O. O., & Sithole, T. (2025). Forecasting retail sales using machine learning models. *American Journal of Statistics and Actuarial Science*. https://ajpojournals.org/journals/AJSAS/article/view/2679

Oncioiu, I., Bunget, O. C., Türkes, M. C., Capusneanu, S., Topor, D. I., Tamas, A. S., Rakos, I. S., & Hint, M. S. (2019). The impact of big data analytics on company performance in supply chain management. *Sustainability, 11*(18), 4864. https://doi.org/10.3390/su11184864

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, R., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., et al. (2022). Forecasting: Theory and practice. *International Journal of Forecasting, 38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001

Phyu, M. M., & Khine, M. T. (2023). Retail demand forecasting using sequence to sequence long short-term memory networks. In *Proceedings of the 2023 IEEE Conference on Computer Applications (ICCA)* (pp. 208–213). IEEE.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0

ToolsGroup. (2026). *Machine learning in demand planning: How to boost forecasting*. https://www.toolsgroup.com/blog/machine-learning-in-demand-planning-how-to-boost-forecasting/

Vukovic, D. B., Spitsina, L., Gribanova, E., Spitsin, V., & Lyzin, I. (2023). Predicting the performance of retail market firms: Regression and machine learning methods. *Mathematics, 11*(8), 1916. https://doi.org/10.3390/math11081916

Weber, F., & Schütte, R. (2019). A domain-oriented analysis of the impact of machine learning—the case of retailing. *Big Data and Cognitive Computing, 3*(1), 11. https://doi.org/10.3390/bdcc3010011

Xie, L., Liu, J., & Wang, W. (2024). Predicting sales and cross-border e-commerce supply chain management using artificial neural networks and the Capuchin search algorithm. *Scientific Reports, 14*, 13340. https://doi.org/10.1038/s41598-024-62368-6

Appendix A. Model Hyperparameters and Software Environment

For reproducibility, Table A1 reports the hyperparameter settings used for each learned model, and the software environment is documented below. All models were implemented in Python 3 using scikit-learn 1.8.0 (Pedregosa et al., 2011); pandas and NumPy were used for panel construction and lag-feature engineering; matplotlib and seaborn were used for all figures. As documented, the XGBoost, TensorFlow, and PyTorch libraries were unavailable in the execution environment and could not be installed due to restricted network access, which is the basis for the substitutions disclosed throughout this paper.

Table A1. Hyperparameter settings for all six forecasting models.

Model	Key Hyperparameters
Linear Regression	Ordinary least squares; no regularisation; intercept fitted
Random Forest	n_estimators = 300; max_depth = 6; random_state = 42; n_jobs = -1
Gradient Boosted Trees (XGBoost-equivalent)	n_estimators = 300; max_depth = 3; learning_rate = 0.05; random_state = 42
ANN (windowed MLP)	hidden_layer_sizes = (64, 32); activation = ReLU; max_iter = 3000; early_stopping = True; random_state = 42
Naive Baseline	Prediction = lag_1 (previous month's revenue for the same category)
Moving Average Baseline	Prediction = 3-month rolling mean of revenue, computed on lagged values only

All continuous features were standardised (zero mean, unit variance) using a scikit-learn StandardScaler fitted exclusively on the training partition, and the categorical product-category feature was one-hot encoded using a scikit-learn OneHotEncoder configured to handle unseen categories gracefully at inference time. The train/test split was strictly chronological to prevent temporal leakage between the two partitions.