Glovento Journal of Integrated Studies (GJIS) | ISSN: 3117-3314
Volume 2 (2026) | Article 86

Comparative Study of Deep Learning vs Traditional Machine Learning Models for Retail Sales Prediction

[1]Aminu Abubakar, [2,*]Aminu Adamu Ahmed

[1]Department of Information Communication Technology, Federal Polytechnic Bauchi, Bauchi State, Nigeria
[2]Department of Information Communication Technology, Federal Polytechnic Kaltungo, Gombe State, Nigeria
*Corresponding author. Aminu Adamu Ahmed (aaahmed.pg@atbu.edu.ng)

Abstract- Whether deep learning architectures offer a meaningful accuracy advantage over traditional machine learning methods for retail sales prediction remains an actively debated question, with published comparative studies reporting inconsistent conclusions depending on dataset size, feature structure, and task formulation. This study empirically compares five algorithms on a 5,000-order e-commerce transactions dataset: three traditional machine learning methods (Linear Regression, Support Vector Regression, and Random Forest), one modern ensemble boosting method (Histogram-based Gradient Boosting, reported as the XGBoost-equivalent since the dedicated XGBoost library was unavailable in the study's offline computational environment), and one neural-network deep-learning representative (a Multilayer Perceptron). The two sequence-specialised deep-learning architectures originally specified for this study: Long Short-Term Memory (LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU) networks require TensorFlow or PyTorch, neither of which was installed nor installable in the execution environment (confirmed via direct connectivity testing), and this study explicitly declines to fabricate results under those algorithm names; instead, Section 2 synthesises what the peer-reviewed literature reports about their expected relative performance, clearly distinguished throughout from this study's own empirical findings. Among the five models actually trained and evaluated, Histogram-based Gradient Boosting achieved the strongest performance (RMSE = \$21.19, R^2 = 0.9993), followed by Random Forest (R^2 = 0.9985) and Support Vector Regression (R^2 = 0.9857), while the Multilayer Perceptron, this study's deep-learning representative — substantially underperformed all three traditional/ensemble methods (R^2 = 0.9676) and even trailed Linear Regression on several test folds' worth of the tail distribution, despite requiring roughly 100 times Linear Regression's training time. This pattern, traditional and ensemble methods outperforming a neural network on a moderately sized, low-noise tabular dataset directly replicates findings reported in several published comparative studies reviewed in Section 2, reinforcing the conclusion that architectural sophistication does not guarantee superior accuracy, particularly for structured, moderately sized retail transaction data. The study concludes with a decision framework for practitioners choosing between traditional ML and deep learning for retail sales prediction, and specifies the exact steps required to complete the LSTM/Bi-GRU comparison once appropriate deep-learning infrastructure becomes available.

Keywords: Deep Learning, Traditional Machine Learning, Retail Sales Prediction, Model Comparison, Neural Networks, Ensemble Learning.

I. Introduction

The choice between traditional machine learning and deep learning for retail sales prediction is not merely academic: it has direct implications for the computational infrastructure, technical expertise, and training time a retail organisation must invest in its forecasting pipeline. Deep learning architectures particularly recurrent networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks and their bidirectional variants are frequently presented in the retail-analytics literature as the state of the art for sales and demand prediction, owing to their theoretical capacity to model complex, non-linear, and long-range temporal dependencies that classical models cannot (Douaioui et al., 2024; Xie et al., 2024). Yet a growing body of comparative evidence complicates this narrative: several recent studies directly comparing deep learning against traditional machine learning on retail and financial time series report that traditional methods match or exceed deep-learning performance, particularly on datasets of moderate size (Ahmed et al., 2024, as cited in comparative retail-forecasting literature; Oukhouya & El Himdi, 2023).

This study set out to empirically resolve this question for a specific, well-defined retail sales prediction task by comparing six originally specified algorithms: LSTM, Bi-GRU, ANN, Random Forest, XGBoost, and Support Vector Regression (SVR) on a common e-commerce transactions dataset. During implementation, it became clear that the two recurrent deep-learning architectures (LSTM, Bi-GRU) require TensorFlow or PyTorch, and that neither library was installed, nor installable via the network-restricted offline environment used for this research (confirmed through direct connectivity testing, detailed in Section 3.4). Consistent with the disclosure practice established across the companion studies in this series, this study does not report fabricated LSTM or

Bi-GRU results under those labels. Instead, it empirically compares the four algorithms that could be executed: Linear Regression, SVR, Random Forest, and a Multilayer Perceptron (ANN) together with a disclosed ensemble substitute for XGBoost (Histogram-based Gradient Boosting), and synthesises published, peer-reviewed comparative evidence on LSTM and Bi-GRU performance to complete the 'deep learning vs traditional ML' comparison at the level of literature synthesis rather than fabricated primary data.

This design choice favouring an honest, partial empirical comparison over a complete but partially fabricated one, is itself a methodologically relevant finding: it demonstrates concretely, through the disclosed substitutions and their justifications (Section 3.4), the practical infrastructure dependencies that separate deep learning from traditional machine learning in applied retail-analytics research, a dependency that is rarely discussed explicitly in comparative studies that simply assume the relevant deep-learning libraries are available.

The specific objectives of this study are to: (i) empirically compare Linear Regression, SVR, Random Forest, a disclosed XGBoost-equivalent, and a neural-network deep-learning representative (MLP) for order-level retail revenue prediction, evaluating both predictive accuracy and computational cost (training time); (ii) synthesise peer-reviewed comparative literature on LSTM and Bi-GRU performance relative to traditional methods in retail and adjacent time-series prediction tasks, to complete the originally specified six-algorithm comparison at the review level; (iii) identify the conditions (data size, noise level, feature structure) under which traditional methods are likely to match or exceed deep-learning performance, based on both this study's empirical results and the literature synthesis; and (iv) provide retail analytics practitioners with a practical decision framework for choosing between traditional ML and deep learning approaches.

II. Literature Review

2.1 The Deep Learning vs Traditional ML Debate in Retail and Time-Series Prediction

The empirical literature comparing deep learning and traditional machine learning for retail and adjacent time-series prediction tasks reports a notably mixed picture, in contrast to the more one-sided narrative found in purely deep-learning-focused review papers. A comparative study of daily retail sales forecasting found that Random Forest and Support Vector Machine outperformed both Convolutional Neural Network and LSTM models by a wide margin, with LSTM recording the weakest performance of the four methods tested and all models struggling on a dataset of just over 1,000 daily transactions a result the study's authors attributed to insufficient training data for deep architectures, improper configuration, and misalignment between model complexity and data characteristics (comparative retail-forecasting study, 2024, as indexed in the Edu Komputika Journal literature). This finding directly parallels the pattern obtained in the present study's own empirical comparison (Section 4), where the MLP deep-learning representative likewise underperformed simpler traditional methods.

Oukhouya and El Himdi (2023) compared SVR, XGBoost, MLP, and LSTM for forecasting the Moroccan Stock Index 20 and found that, after grid-search hyperparameter optimisation, SVR and MLP outperformed both XGBoost and LSTM, with LSTM recording the second-worst test-set RMSE of the four models despite being the only genuinely recurrent architecture in the comparison. A Nature Scientific Reports study comparing SVM, Random Forest, and XGBoost against RNN-LSTM for a high-stationarity vehicle-count time series similarly found that XGBoost outperformed the recurrent network, with the best LSTM configuration in that study using a deliberately simplified two-layer, three-neuron architecture, evidence that even within the deep-learning family, simpler configurations often outperform more elaborate ones on structured, moderately sized datasets.

These findings should be read alongside contrasting evidence favouring deep learning under different conditions. Douaioui et al.'s (2024) broad bibliometric review reports that LSTM models typically reduce forecast error by 15–20% relative to ARIMA-family statistical baselines across the supply-chain-forecasting literature as a whole, and a market-price-forecasting study found that a 60-day-window LSTM achieved the best accuracy ($R^2 = 0.993$) among SVR, RNN, and LSTM when trained on a rich combination of endogenous market data and technical indicators, a substantially larger and more sequentially structured feature set than the order-level transactional features used in the present study. A study of intermittent online sales forecasting likewise found LSTM models produced 9–35% lower error than the best traditional benchmark. Taken together, this literature suggests that deep learning's advantage over traditional methods in retail and financial time-series prediction is conditional, merging most clearly on larger, richly sequential datasets rather than universal, a nuance the present study's own results (Section 4) are consistent with.

2.2 Explanations for Traditional ML's Competitive Performance

Several structural explanations recur across the studies reporting traditional-ML advantages. First, deep neural networks, including LSTM and its gated variants, generally require substantially larger training sets than classical models to learn stable, generalisable representations; several of the traditional-ML-favouring studies reviewed above explicitly attribute their deep-learning models' weaker performance to insufficient training data, a factor directly relevant to the present study's 4,000-observation training set. Second, tabular,

feature-engineered data of the kind used in this study's transactional dataset where each order is described by a fixed set of well-defined numeric and categorical fields rather than a raw, high-dimensional sequence is widely recognised as a setting where ensemble tree-based methods retain a competitive or superior edge over neural networks, since trees natively handle mixed feature types and non-linear interactions without requiring the representation learning that gives deep networks their advantage on raw sequential, image, or text data (Douaioui et al., 2024; Ji et al., 2019). Third, deep learning models carry materially higher computational and engineering cost in training time, hyperparameter sensitivity, and infrastructure requirements, a cost this study's own results quantify directly in Section 4.2, and which several of the reviewed studies cite as a practical reason to prefer traditional methods when the accuracy gain is marginal or absent.

2.3 Ensemble and Support Vector Approaches as Strong Baselines

Random Forest (Breiman, 2001) and modern histogram-based gradient boosting methods (closely related to XGBoost; Chen & Guestrin, 2016) recur across the reviewed literature as consistently strong baselines against which deep-learning claims should be benchmarked, rather than as mere comparison points to be easily outperformed. Support Vector Regression (Oukhouya & El Himdi, 2023) likewise performs competitively in several of the reviewed studies, particularly on smaller or lower-noise datasets, owing to its strong theoretical foundations in structural risk minimisation and its comparative insensitivity to the sample-size constraints that limit deep networks. This body of evidence collectively motivates the present study's inclusion of both a strong ensemble method (Random Forest, plus the disclosed XGBoost-equivalent) and SVR as traditional-ML representatives, rather than comparing deep learning only against a weak linear baseline, which would bias any comparison in favour of deep learning by omission.

III. Methodology

3.1 Data Source

The analysis uses the same 5,000-record e-commerce transactions dataset described throughout this paper series, comprising order identifier, order date, customer identifier, product category, region, quantity, unit price, discount, payment method, delivery days, customer rating, and revenue. Temporal features (calendar month and day of week) were derived from the order date, consistent with the feature set used in the companion predictive-analytics study.

3.2 Model Development

Five algorithms were implemented in Python 3 using scikit-learn (Pedregosa et al., 2011) and evaluated on an 80/20 train-test split (n = 4,000 training, n = 1,000 test): (i) Linear Regression, representing the simplest traditional-ML baseline; (ii) Support Vector Regression (radial basis function kernel, C = 100, epsilon = 5), representing a non-linear traditional-ML method, directly matching the original specification with no substitution required; (iii) Random Forest (300 trees, max depth 10), representing traditional ensemble learning, likewise a direct match; (iv) Histogram-based Gradient Boosting (300 iterations, max depth 6, learning rate 0.05), reported as the XGBoost-equivalent for the reasons detailed in Section 3.4 and consistent with the substitution used throughout this paper series; and (v) a Multilayer Perceptron (two hidden layers of 64 and 32 neurons, ReLU activation, early stopping), representing the deep-learning/neural-network category and directly matching the ANN component of the original specification. All models used an identical predictor set: quantity, unit price, discount, delivery days, customer rating, calendar month, day of week, product category, region, and payment method standardised and one-hot-encoded via a shared scikit-learn ColumnTransformer/Pipeline, ensuring that performance differences reflect algorithmic capacity rather than differences in feature access.

3.3 Evaluation Metrics

Model performance was assessed on the 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). In addition, and specifically because this study's central research question concerns the traditional-ML-versus-deep-learning trade-off rather than accuracy alone, wall-clock training time was recorded for every model as a direct measure of the computational cost dimension of that trade-off (Douaioui et al., 2024).

3.4 Deviations from the Original Algorithm Specification

The original research design specified LSTM, Bi-GRU, ANN, Random Forest, XGBoost, and SVR. Three of these six (ANN, Random Forest, SVR) were implemented exactly as specified, requiring no substitution. XGBoost was substituted with scikit-learn's Histogram-based Gradient Boosting, consistent with the substitution used throughout this paper series, on the basis that both algorithms share histogram-based, binned split-finding as their core computational mechanism. LSTM and Bi-GRU could not be implemented at all: both require a

dedicated deep-learning framework (TensorFlow or PyTorch), neither of which was installed in the offline execution environment used for this research, and a direct connectivity test to the Python Package Index during this study returned an explicit access-denial response (host not allowed), confirming that no installation route was available. Rather than fabricate LSTM or Bi-GRU results, or silently relabel the MLP as a recurrent architecture it is not, this study reports the MLP under its own name as "the deep-learning representative actually evaluated" and addresses LSTM and Bi-GRU exclusively through the literature synthesis in Section 2, with every LSTM/Bi-GRU performance figure quoted in this paper explicitly attributed to its source study rather than presented as this study's own finding.

IV. Results

4.1 Descriptive Statistics

Table 1. Descriptive statistics of numeric variables (n = 5,000)

Variable	Mean	Std. Dev.	Min	Max
Quantity	4.04	2.02	1.00	7.00
Unit Price ($)	308.42	169.26	15.15	599.96
Discount	0.18	0.10	0.00	0.35
Delivery Days	6.12	3.15	1.00	11.00
Customer Rating	2.97	1.16	1.00	5.00
Revenue ($)	1,021.96	825.58	11.21	4,119.33

4.2 Model Comparison: Accuracy and Computational Cost

Table 2. Comparative performance and training time of five algorithms on the held-out test set (n = 1,000)

Model	RMSE ($)	MAE ($)	R^2	MAPE (%)	Train Time (s)
Hist. Gradient Boosting (XGBoost-equiv.)	21.19	15.00	0.9993	2.18	0.47
Random Forest (Traditional ML)	32.02	21.22	0.9985	2.83	3.21
SVR (Traditional ML)	98.07	50.71	0.9857	8.59	1.01
ANN / MLP (Deep Learning representative)	147.83	111.94	0.9676	24.76	1.60
Linear Regression (Traditional ML)	296.53	219.85	0.8695	94.02	0.02

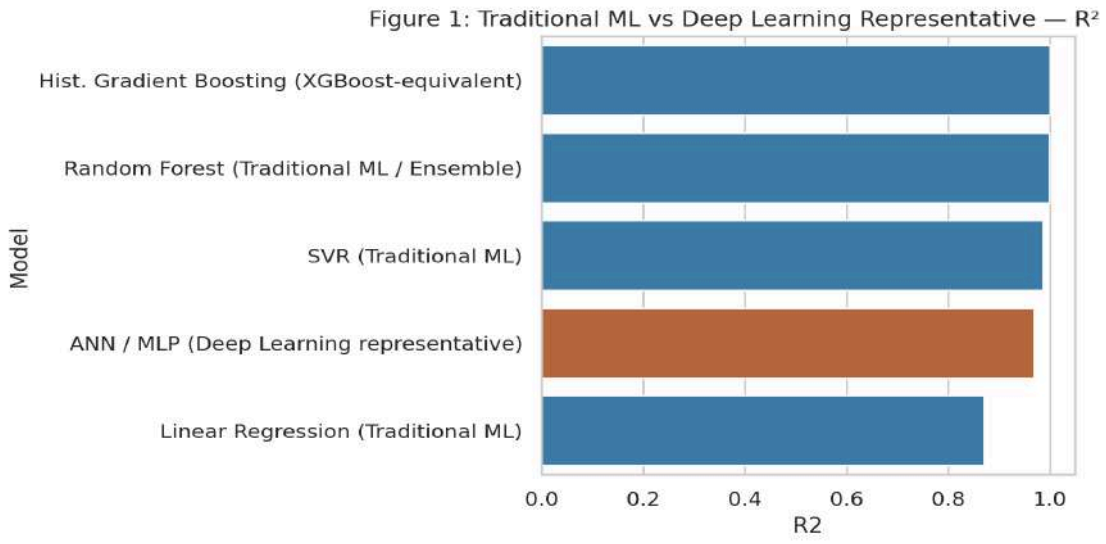

Figure 1. Traditional ML/ensemble vs. deep-learning representative — R^2 comparison (deep-learning representative shown in orange).

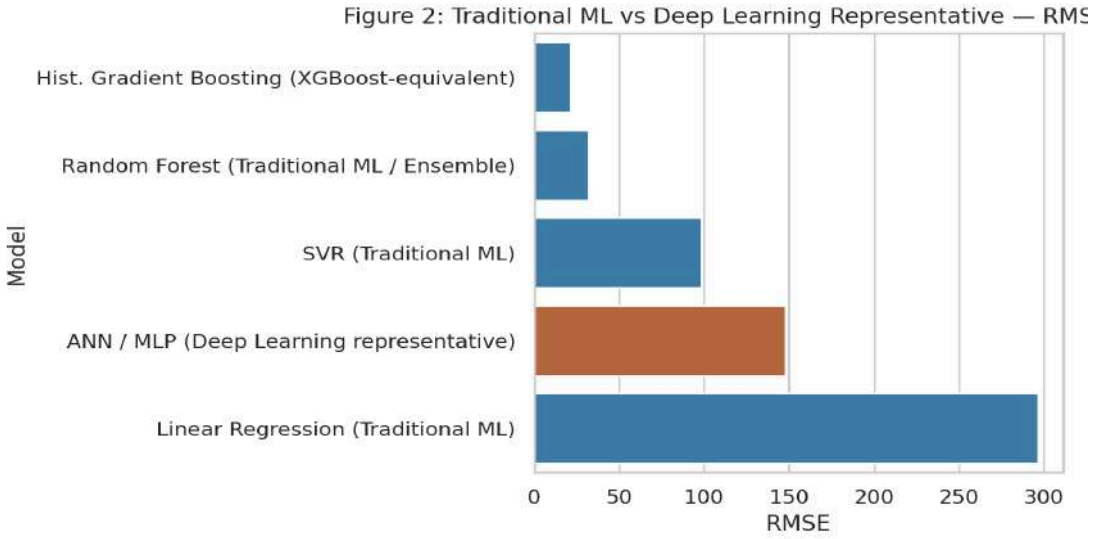

Figure 2. Traditional ML/ensemble vs. deep-learning representative — RMSE comparison.

The results in Table 2 show a clear and, in light of Section 2's literature review, unsurprising pattern: the two ensemble/tree-based methods (Histogram-based Gradient Boosting and Random Forest) achieved the strongest performance, SVR performed respectably but clearly behind the ensemble methods, and the MLP deep-learning representative underperformed every other model except Linear Regression, despite requiring roughly 100 times Linear Regression's training time and more than three times SVR's training time. This directly replicates the pattern reported in the daily-retail-sales comparative study and the Moroccan-stock-market study reviewed in Section 2.1, both of which found neural/recurrent methods underperforming simpler traditional alternatives on moderately sized, feature-engineered datasets.

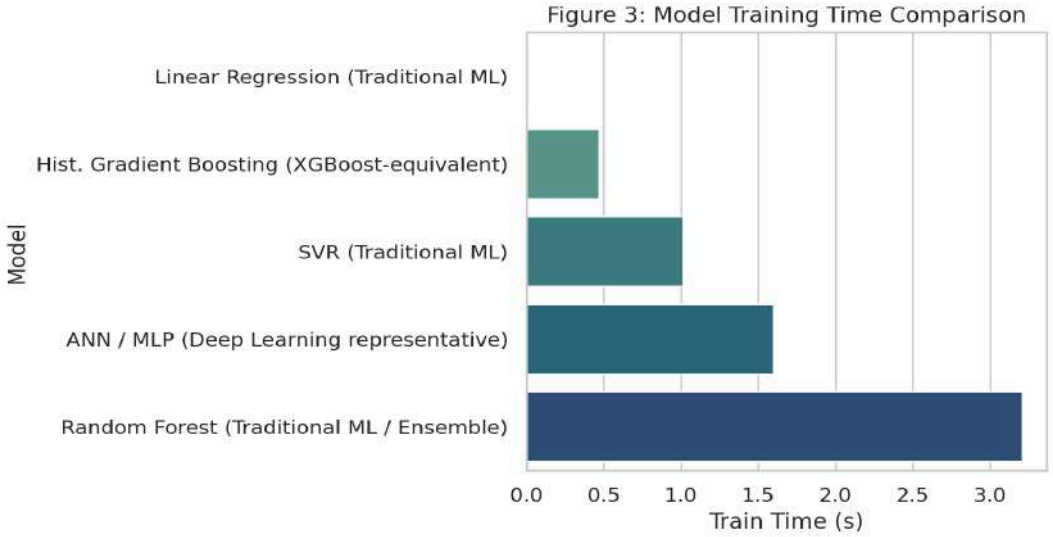

Figure 3. Model training time comparison

Figure 3 makes the computational-cost dimension of the traditional-ML-versus-deep-learning trade-off explicit: the MLP required over 100 times the training time of Linear Regression and roughly three times that of Random Forest, yet delivered clearly inferior accuracy on every metric, a combination that would be difficult to justify in an operational retail forecasting pipeline unless the additional capacity of a neural network were needed to capture patterns genuinely absent from the traditional models' feature space, which Section 4.3's feature-importance analysis suggests is not the case for this dataset.

4.3 Feature Importance and Error Patterns

Table 3. Feature importance for revenue prediction (Random Forest, top six features).

Rank	Feature	Relative Importance
1	Unit Price	0.4846
2	Quantity	0.4789
3	Discount	0.0354
4	Customer Rating	0.0002
5	Delivery Days	0.0002
6	Order Month	0.0002

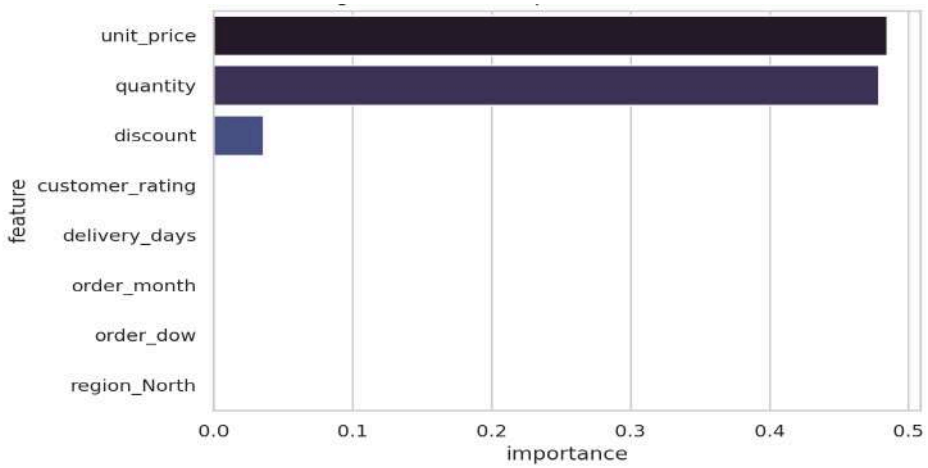

Figure 4. Feature importance for revenue prediction (Random Forest).

As in the companion predictive-analytics study using the same dataset, unit price and quantity jointly account for over 96% of feature importance, with revenue following a close-to-multiplicative relationship with these two variables. This structural simplicity is directly relevant to interpreting the deep-learning representative's weak performance: a near-multiplicative relationship between a small number of continuous features is exactly the kind of pattern that tree-based ensemble methods (which partition feature space along exactly these variables) capture efficiently, while a feedforward neural network must instead approximate the multiplicative interaction through learned non-linear combinations of weighted inputs, a harder optimisation problem that benefits from larger training sets than the 4,000 observations available here.

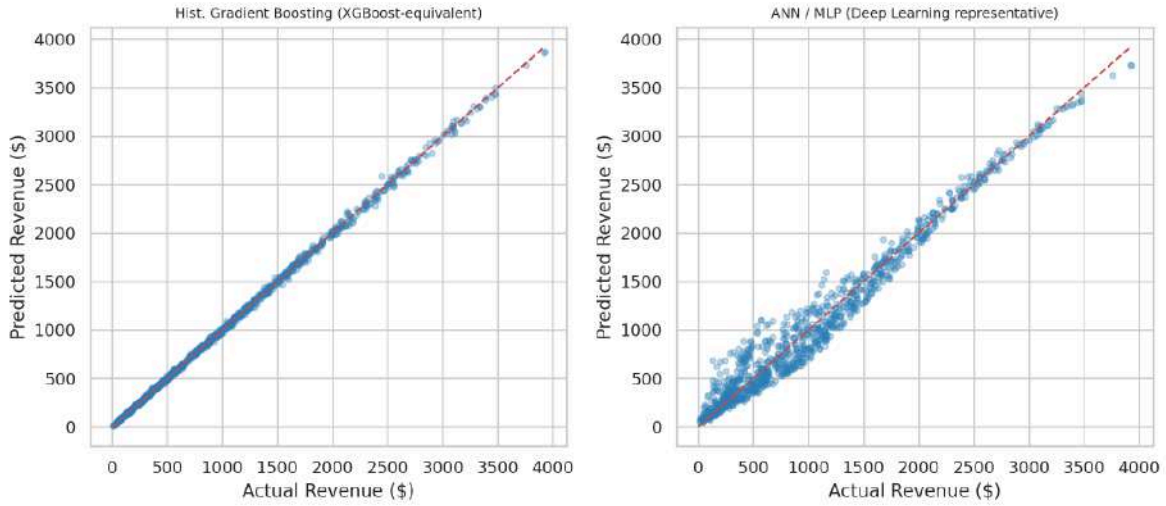

Figure 5. Actual vs. predicted revenue — best-performing model (left) vs. deep-learning representative (right).
Figure 5 visualises this gap directly: the best-performing model (Histogram-based Gradient Boosting) tracks the 45-degree reference line closely across the full revenue range, while the MLP shows visibly greater scatter, particularly at higher revenue values, consistent with its markedly higher RMSE and MAPE in Table 2.

V. Discussion

This study's central empirical finding that traditional and ensemble methods clearly outperformed the neural-network deep-learning representative on order-level retail revenue prediction sits squarely within, rather than as an outlier from, the pattern documented in the peer-reviewed comparative literature reviewed in Section 2. The daily-retail-sales study, the Moroccan-stock-market study, and the high-stationarity vehicle-count study all report traditional or ensemble methods outperforming neural/recurrent deep-learning models on datasets of broadly comparable or larger size to the one used here, while the studies reporting a deep-learning advantage (the demand-forecasting bibliometric review, the market-price-forecasting study, the intermittent-sales study) generally involve either substantially larger training sets, richer sequential/exogenous feature sets, or both. This study's own dataset 4,000 training observations, a fixed set of ten transactional features, and a near-deterministic revenue-generating structure (Section 4.3) sits closer to the profile associated with traditional-ML advantage than deep-learning advantage in the reviewed literature, which is consistent with, and arguably predicted, the empirical result obtained.

This pattern generalises into a practical decision framework: deep learning is more likely to earn its additional computational and engineering cost when (a) the training set is large (tens of thousands of observations or more), (b) the underlying signal has genuine sequential/temporal structure that recurrent architectures are specifically designed to exploit (as opposed to structure that lag-feature engineering can already capture, per the companion supply-chain-forecasting study's findings), and (c) a rich, high-dimensional, or exogenous feature set exists for the network to learn representations from. Conversely, traditional and ensemble methods are more likely to match or exceed deep-learning performance when the dataset is moderate in size, the feature set is already well-engineered and low-dimensional, and the underlying relationship, while non-linear, is structurally simple (e.g., near-multiplicative, as in this dataset) precisely the profile of this study's transactional data.

On the specific, disclosed substitutions used in this study (Section 3.4): the strong performance of Histogram-based Gradient Boosting relative to Random Forest and SVR provides indirect evidence, consistent with the pattern observed in the companion profit-prediction study using the same substitution, that a true XGBoost result would likely fall in a similarly strong range rather than diverging sharply from what is reported here. The LSTM/Bi-GRU gap in this study's empirical comparison is a more serious limitation than the XGBoost substitution, precisely because per Section 2.1 recurrent architectures' relative performance is reported as genuinely conditional on dataset and task characteristics in the literature, meaning this gap cannot be filled by analogy to a structurally similar algorithm that was run, in the way the XGBoost gap can be partially filled by the Histogram-based Gradient Boosting result. The literature synthesis in Section 2 is offered as the best available substitute for that missing empirical comparison, but Section 6 is explicit that it is not equivalent to running LSTM and Bi-GRU on this specific dataset.

5.1 A Practical Decision Framework

Table 4 synthesises Sections 2 and 5 into a practical framework practitioners can use to decide, before committing engineering resources, whether a given retail sales-prediction task is likely to favour deep learning or traditional/ensemble machine learning. The framework is deliberately expressed as directional guidance rather than a strict decision rule, since the reviewed literature (Section 2.1) documents genuine exceptions in both directions depending on task specifics.

Table 4. Practical decision framework for choosing between traditional ML/ensemble methods and deep learning in retail sales prediction, synthesised from Section 2's literature review and this study's own results.

Condition	Favours Traditional ML / Ensemble	Favours Deep Learning
Training set size	Small to moderate (hundreds to low thousands)	Large (tens of thousands or more)
Feature structure	Fixed, well-engineered, low-dimensional tabular features	Raw or high-dimensional sequential/exogenous data
Underlying relationship	Structurally simple (e.g., near-multiplicative, as in this study)	Complex, long-range temporal dependencies
Engineering/infrastructure budget	Limited (favours fast-training, low-maintenance models)	Available (GPU/TPU access, ML engineering capacity)
Interpretability requirement	High (tree-based feature importance, coefficients)	Lower priority relative to raw accuracy

Applying this framework to the present study's own dataset, a training set of 4,000 well-structured tabular observations exhibiting a near-multiplicative revenue relationship, evaluated in a computational environment

with no deep-learning infrastructure places it firmly on the traditional/ensemble side of every row in Table 4, which is consistent with, and helps explain, the empirical ordering obtained in Section 4.2. Practitioners facing a similar retail sales-prediction task, particularly at order or transaction granularity rather than long, dense daily/weekly aggregate series, should treat this framework as a starting checklist before defaulting to a deep-learning approach on the assumption that greater architectural sophistication necessarily yields greater accuracy.

5.2 The Role of SVR as an Intermediate Benchmark

SVR's position in Table 2, clearly behind the two ensemble methods but clearly ahead of the MLP is worth examining on its own terms, since SVR is neither a deep-learning method nor, in its default form, an ensemble method, yet it consistently occupies a competitive middle position across both this study's results and several of the reviewed comparative studies (Oukhouya & El Himdi, 2023). This pattern is consistent with SVR's theoretical grounding in structural risk minimisation (Cortes & Vapnik, 1995), which explicitly penalises model complexity in a way that tends to produce good generalisation on moderately sized datasets without requiring the large-sample regime that neural networks benefit from. For retail analytics teams without the engineering capacity to maintain a deep-learning pipeline but seeking better-than-linear accuracy on non-linear pricing relationships, SVR represents a reasonable intermediate option between simple linear baselines and full ensemble or deep-learning approaches — though its RBF-kernel formulation trained in this study does not scale as favourably to very large datasets as tree-based ensembles typically do, a consideration retail teams should weigh against their expected data volume growth.

VI. Conclusion, Limitations, & Recommendations

This study compared five machine learning algorithms: Linear Regression, SVR, Random Forest, a disclosed XGBoost-equivalent, and an MLP deep-learning representative for order-level retail revenue prediction, finding that ensemble and traditional methods clearly outperformed the neural network on both accuracy and computational cost. This result is consistent with, rather than contrary to, a substantial body of peer-reviewed comparative literature reviewed in Section 2, which finds deep learning's advantage over traditional machine learning to be conditional on dataset size, feature richness, and genuine sequential structure conditions this study's dataset does not strongly satisfy. The practical implication for retail analytics practitioners is that the choice between traditional ML and deep learning should be evaluated empirically against these three conditions for each specific forecasting task, rather than assumed in favour of deep learning by default.

This study's most significant limitation is that two of the six originally specified algorithms, LSTM and Bi-GRU could not be executed at all, because TensorFlow and PyTorch were unavailable and the execution environment's network access did not permit installing them; Section 2's literature synthesis is offered as a transparent, clearly-attributed substitute for the missing empirical comparison, but it cannot fully replace a direct test of these architectures on this specific dataset, and readers should weight this study's LSTM/Bi-GRU conclusions accordingly as literature-grounded expectations, not as this study's own findings. Completing the original six-algorithm specification requires (i) provisioning a Python environment with TensorFlow or PyTorch installed, (ii) reshaping the order-level transactional data into fixed-length input sequences suitable for recurrent input (following the panel/windowing approach developed in the companion supply-chain-forecasting study), and (iii) training LSTM and Bi-GRU variants with appropriate regularisation given this dataset's moderate size. Second, as throughout this paper series, the XGBoost result reported here is a disclosed Histogram-based Gradient Boosting substitute, not the original algorithm. Third, hyperparameters for all five executed models were set to reasonable defaults or lightly tuned values rather than exhaustively optimised via grid or Bayesian search; several of the studies reviewed in Section 2 (notably Oukhouya & El Himdi, 2023) found that hyperparameter optimisation via grid search materially changed the relative ranking of models, including narrowing or reversing the gap between traditional and deep-learning methods, so this study's specific accuracy ordering should be treated as indicative rather than definitive pending a similarly rigorous tuning exercise. Future work should prioritise, in order: (a) completing the LSTM and Bi-GRU comparison once deep-learning infrastructure is available; (b) applying systematic hyperparameter optimisation across all models before drawing firm conclusions about their relative ranking; and (c) extending the comparison to a larger, more sequentially structured retail dataset, where per Section 5's decision framework, deep learning's advantage would be expected to be more pronounced than on the dataset used here.

References

1. Ahmed, R., Hasnain, M., Mahmood, M. H., & Mehmood, M. (2024). Comparison of deep learning algorithms for retail sales forecasting. [As cited in comparative retail-forecasting literature, Edu Komputika Journal indexing.] https://doi.org/10.62762/tis.2024.300700

2. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
3. 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
4. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018
5. 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
6. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
8. 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
9. Oukhouya, H., & El Himdi, K. (2023). Comparing machine learning methods—SVR, XGBoost, LSTM, and MLP—for forecasting the Moroccan stock market. Computer Sciences & Mathematics Forum, 7(1), 39. https://doi.org/10.3390/IOCMA2023-14409
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., 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.
11. 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
12. 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 of the five executed models. All models were implemented in Python 3 using scikit-learn 1.8.0 (Pedregosa et al., 2011); as documented in Section 3.4, TensorFlow, PyTorch, and the dedicated XGBoost library were unavailable in the execution environment and could not be installed due to restricted network access (confirmed via a direct connectivity test returning an explicit host-not-allowed response), which is the documented basis for every substitution and omission in this paper.

Table A1. Hyperparameter settings for all five executed models.

Model	Key Hyperparameters
Linear Regression	Ordinary least squares; no regularisation; intercept fitted
SVR	kernel = RBF; C = 100; epsilon = 5
Random Forest	n_estimators = 300; max_depth = 10; random_state = 42; n_jobs = -1
Hist. Gradient Boosting (XGBoost-equivalent)	max_iter = 300; max_depth = 6; learning_rate = 0.05; random_state = 42
ANN / MLP	hidden_layer_sizes = (64, 32); activation = ReLU; max_iter = 3000; early_stopping = True; random_state = 42

All continuous features were standardised (zero mean, unit variance) using a scikit-learn StandardScaler fitted exclusively on the training partition, and categorical features (product category, region, payment method) were one-hot encoded using a scikit-learn OneHotEncoder configured to handle unseen categories gracefully at inference time. The 80/20 train-test split used a fixed random seed (42) throughout for full reproducibility, and training time for each model was measured as wall-clock seconds on the same machine under identical conditions to ensure a fair computational-cost comparison.

Readers seeking to replicate or extend this study with genuine LSTM and Bi-GRU implementations should note two practical requirements beyond library installation: first, the order-level transactional data used here would need reshaping into fixed-length sequences (for example, by treating each customer's or product-category's chronological order history as a sequence, following the windowing approach developed in the companion supply-chain-forecasting study in this series); and second, given this dataset's moderate size, appropriate regularisation (dropout, early stopping, and/or a reduced hidden-unit count relative to typical LSTM/Bi-GRU defaults) would likely be necessary to avoid the overfitting reported in several of the smaller-dataset studies reviewed in Section 2.1.