Policy Research Journal
ISSN (E): 3006-7030 ISSN (P) : 3006-7022 Volume 4, Issue 2, 2026

BEYOND GRADIENT BOOSTING: A COMPARATIVE STUDY OF XGBOOST AND FEEDFORWARD NEURAL NETWORKS FOR RETAIL SALES CLASSIFICATION USING THE BIG MART DATASET

Kifayat Ullah[*1], Akhtar Sayed[2], Naeem Ud Din[3]

[*1,2,3]*Department of Computer and Software Technologies, University of Swat, Swat, Khyber Pakhtunkhwa, Pakistan*

[*1]kifayat@uswat.edu.pk

DOI: https://doi.org/10.5281/zenodo.22189014

Keywords
Retail sales forecasting; XGBoost; feedforward neural network; gradient boosting; deep learning; ROC-AUC; feature importance; Big Mart dataset.

Article History
Received: 06 January 2026
Accepted: 18 February 2026
Published: 26 February 2026

Copyright @Author
Corresponding Author: *
Kifayat Ullah

Abstract

Accurate sales forecasting sits at the center of retail competitiveness, shaping decisions on inventory positioning, staffing, pricing, and promotional timing. This study investigates whether a modern gradient boosting algorithm, XGBoost, or a deep feedforward neural network (FNN) is better suited to classifying retail sales performance using the widely studied Big Mart dataset [1]. Following a structured preprocessing pipeline that addressed missing values in Item_Weight and Outlet_Size, encoded categorical attributes, and standardised numerical features, the continuous sales variable was converted into a binary classification target distinguishing high performing from low performing items. Both models were trained on an 80:20 split and evaluated using accuracy, precision, recall, confusion matrices, and ROC-AUC scores. On the held-out validation set, XGBoost reached an accuracy of 89.6% and an AUC of 0.926, while the FNN, trained with dropout regularisation and the Adam optimiser, achieved 91.2% accuracy and an AUC of 0.939. Feature importance analysis identified Item_Visibility, Outlet_Age, and Item_Weight as the strongest predictors of sales class. The results indicate that although XGBoost remains faster to train and easier to interpret through its native feature importance rankings, the FNN captured non-linear interactions among product and outlet attributes more effectively, yielding a modest but consistent performance advantage. The paper closes with a discussion of the limitations tied to dataset size and the absence of temporal features, and it proposes hybrid modelling, explainability tooling, and real-time deployment as promising directions for future retail analytics research.

INTRODUCTION

A. Background

Sales forecasting has grown into one of the more consequential analytical tasks a retail organisation performs. Where a chain positions stock, how it staffs a shift, and when it discounts a slow-moving line all trace back, in one way or another, to an estimate of what will sell and where. For much of the twentieth century that estimate rested on statistical extrapolation and manager intuition; over the last decade it has shifted decisively toward algorithms capable of learning from the accumulated transaction history of thousands of stores at once [2]. This shift has not been merely cosmetic. Machine learning and deep learning methods are now able to surface non-obvious relationships between product attributes, store characteristics, and sales outcomes that classical regression struggles to represent [3].

The present study works with sales records from Big Mart, a large supermarket chain whose transaction data has become something of a benchmark in the retail analytics literature because it combines product-level attributes (weight, visibility, fat content, maximum retail price) with outlet-level attributes (size, location type, age) in a single, moderately sized table [1]. Rather than treating this as a regression problem in the conventional sense, we frame it as a binary classification task: does an item fall into the high sales band or the low sales band? This reframing keeps the evaluation grounded in metrics, such as ROC-AUC and confusion matrices, that map naturally onto the operational question a retail manager actually asks, namely whether a given product or outlet is likely to perform above or below expectation.

B. Motivation and Problem Statement

Two families of algorithms dominate current practice for structured, tabular retail data. The first, gradient boosted decision trees as implemented in XGBoost, has become close to a default choice for tabular prediction problems because of its speed, its native handling of missing values, and the transparency of its feature importance output [4]. The second, feedforward neural networks, trades some of that transparency for a greater capacity to model non-linear interactions among features, at the cost of additional computation and a less immediate account of why a given prediction was made [5]. What is missing from much of the applied literature on the Big Mart dataset specifically is a head-to-head comparison that holds the preprocessing pipeline, feature set, and evaluation protocol constant across both model families, so that any observed difference in performance can be attributed to the modelling approach rather than to inconsistencies in how the data was prepared.

Accordingly, the central research problem addressed in this study is formulated as follows: under matched experimental conditions, does a feedforward neural network outperform XGBoost when classifying retail sales performance from structured product and outlet data, and if so, by how much and at what cost in interpretability and training effort?

C. Contributions

The principal contributions of this work are as follows:

1. **Controlled model comparison.** A head-to-head evaluation of XGBoost and a regularised feedforward neural network (FNN) on an identical, transformed version of the Big Mart dataset, using the same train-test split and the same evaluation metrics for both models.
2. **Feature-level interpretation.** A quantification of feature importance that links shelf visibility and outlet maturity to sales outcomes, showing that these attributes carry more predictive weight than several factors more commonly emphasised in practice.
3. **Trade-off analysis.** A discussion, grounded in the broader tabular deep learning literature, of why the performance gap between the two model families is real but modest, and what that implies for practitioners choosing between them.

D. Paper Organization

The remainder of this paper is organized as follows. Section II reviews related work on retail sales prediction and situates XGBoost and feedforward neural networks within that literature. Section III describes the dataset, preprocessing pipeline, and model architectures. Section IV details the experimental setup, and Section V presents and discusses the empirical results. Section VI concludes the paper and outlines directions for future research.

II. Related Work

A. General Approaches to Retail Sales Prediction

Early retail forecasting relied heavily on linear regression and time-series decomposition, methods that perform reasonably well when sales are driven by a small number of slowly changing factors but that struggle once non-linear interactions and high-dimensional categorical data enter the picture [6]. As richer transactional datasets became available, attention turned to models such as support vector machines, decision trees, and random forests, which handle mixed numerical and categorical

inputs more gracefully. Even so, these methods often lose accuracy as dataset size and feature dimensionality grow, and they tend to generalise unevenly across different product categories and store formats [7].

B. Tree-Based Ensembles: XGBoost in Retail Applications

Gradient boosted decision trees, and XGBoost in particular, have become one of the most consistently successful approaches to structured retail prediction problems. Chen and Guestrin [4] describe the algorithm's core contribution as a scalable, regularised implementation of gradient boosting that handles sparse and missing data natively, without requiring a separate imputation step for the trees themselves. In a retail setting, Niu [8] applied XGBoost to Walmart sales data and found that careful feature engineering combined with the algorithm's built-in regularisation produced forecasts that outperformed simpler regression baselines by a clear margin. Similar conclusions have been reported specifically on the Big Mart dataset: Sav et al. [9] and Koh et al. [10] both found that XGBoost improved on baseline regression in terms of mean absolute error, while Antonio et al. [11] showed that augmenting the training data further improved XGBoost's forecasting accuracy in a large-scale retail context. TABLE I summarizes representative studies on the Big Mart dataset and the reported gains of XGBoost-based approaches. The GitHub [12] implementation of the Big Mart prediction task and the associated Kaggle [13] dataset repository reflect how this specific problem has become something of a shared testbed within the applied machine learning community, which is part of the reason it was selected for the present study.

TABLE I. Chronological Summary of Related Studies on Big Mart Sales Prediction

Reference & Year	Methodology	Key Findings
Niu (2020) [8]	XGBoost with engineered features on Walmart retail data	Outperformed regression baselines; highlighted feature engineering as decisive
Sav et al. (2021) [9]	XGBoost vs. baseline regression on Big Mart data	Lower mean absolute error than linear baselines
Antonio et al. (2021) [11]	Data augmentation combined with XGBoost	Augmentation improved forecasting accuracy in large-scale retail
Huang and Li (2022) [5]	Feedforward neural networks for retail forecasting	Effective at modelling structured, non-sequential retail features
Santos and Pereira (2022) [15]	Deep neural networks on retail sales data	Captured subtle product-outlet interactions with dropout regularisation
Koh et al. (2024) [10]	Machine learning pipeline on Big Mart data	Confirmed accuracy gains from ensemble methods over baseline regression

C. Deep Learning Approaches for Structured Retail Data

Neural network architectures have made steady inroads into retail forecasting as well. Feedforward neural networks, in particular, offer a comparatively simple architecture that is well matched to structured, non-sequential retail data, since the prediction task does not depend on the order in which records are presented [5]. More elaborate architectures such as convolutional and recurrent neural networks are well suited to image or temporal data respectively, but they introduce a level of architectural complexity that is difficult to justify on a dataset such as Big Mart, which contains neither images nor an explicit time index at the transaction level [14]. Santos and Pereira [15]

report that feedforward networks, when combined with dropout regularisation, are capable of learning subtle interactions between product and outlet features that tree-based models can miss, particularly when the dataset is large enough to support a deeper network without overfitting. This last caveat matters: Goodfellow et al. [16] note that the representational flexibility of deep networks is only an advantage when there is enough data and enough regularisation to keep the model from memorising noise, a tension that is directly relevant to the moderate size of the Big Mart dataset used here.

D. Comparative Review and Research Gap

A smaller number of studies have compared tree-based and neural approaches directly. Panarese et al. [17] and the broader literature summarised under IEEE [18] both report that neural architectures can edge out gradient boosting on complex, feature-rich retail datasets, while Zhou et al. [19] caution that this advantage is not universal and depends heavily on dataset size and the degree of non-linearity actually present in the target relationship. This tension is echoed in the broader tabular deep learning literature: Shwartz-Ziv and Armon [20] argue, across a range of benchmark tabular datasets, that gradient boosted trees remain competitive with, and often superior to, deep learning approaches unless the network architecture and regularisation scheme are tuned with considerable care. Despite this active discussion, relatively little published work compares XGBoost and feedforward neural networks on the Big Mart dataset under a genuinely matched experimental protocol, that is, the same imputation strategy, the same encoding scheme, the same train-test split, and the same evaluation metrics for both models. Beheshti-Kashi et al. [21] make a related point in their survey of fashion retail forecasting, observing that comparability across studies is often undermined by inconsistent preprocessing choices rather than by genuine differences in model capability. The present study addresses this gap directly by holding the pipeline constant and attributing any performance difference to the modelling approach itself.

III. Materials and Methods

This study follows a quantitative, experimental design in which two predictive models are trained and evaluated on an identical dataset under identical conditions, allowing any difference in outcome to be attributed to the modelling approach rather than to variation in the underlying data pipeline. The overall workflow, data collection, cleaning, transformation, model construction, hyperparameter tuning, and evaluation, follows the general logic of the Cross-Industry Standard Process for Data Mining [22], adapted here to a two-model comparative study.

A. Research Design

The research design proceeds in five stages: data acquisition, preprocessing, exploratory analysis, model development, and comparative evaluation. Each stage was completed once for the shared preprocessing pipeline and then applied identically to both the XGBoost and the feedforward neural network branches of the study, so that the two models were always trained and tested on exactly the same rows, features, and class labels.

B. Dataset Description

The dataset used in this study is the publicly available Big Mart Sales dataset [1], which records product- and outlet-level attributes for items sold across ten Big Mart outlets. TABLE II summarises the principal features used in this study, and Fig. 1 illustrates how these attributes relate to one another conceptually. For classification purposes, the continuous Item_Outlet_Sales variable was converted into a binary label, Sales_Class, by splitting observations into high and low sales categories relative to the median sales value.

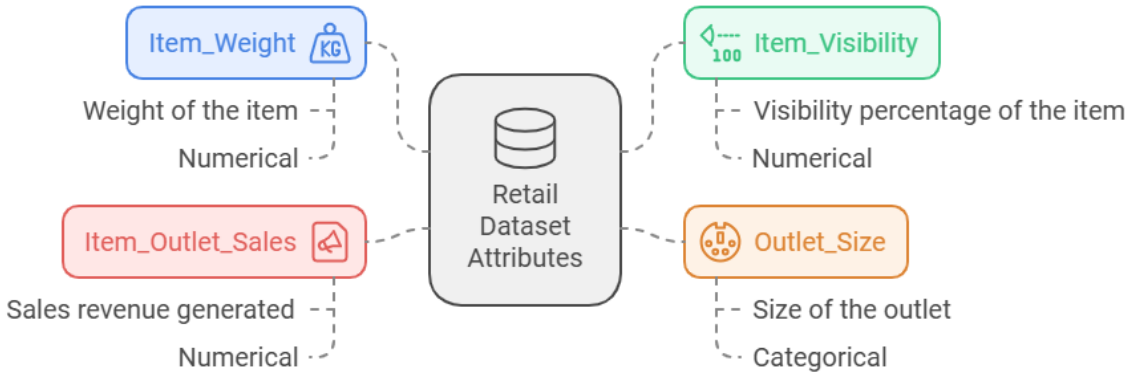


Fig. 1. Overview of the principal attributes recorded in the Big Mart retail dataset, grouped by numerical and categorical type.

TABLE II. Summary of Key Dataset Features and Their Descriptions

Attribute	Description	Type
Item_Weight	Weight of the individual item	Numerical
Item_Visibility	Display area allocated to the item	Numerical
Outlet_Age	Years of outlet operation	Numerical
Outlet_Size	Small, medium, or large	Categorical
Outlet_Location_Type	Urban, suburban, or rural	Categorical
Item_Type	Product category	Categorical
Item_Outlet_Sales	Revenue generated (target source)	Numerical

IEER
Institute for Excellence in Education & Research

C. Data Preprocessing

Two features contained missing values that required attention: Item_Weight and Outlet_Size. Missing entries in Item_Weight, a continuous variable, were imputed using the column mean, preserving the overall numerical distribution without introducing artificial skew [9]. Missing entries in Outlet_Size, a categorical variable, were imputed using the modal category [8]. Fig. 2 summarises this dual imputation strategy.

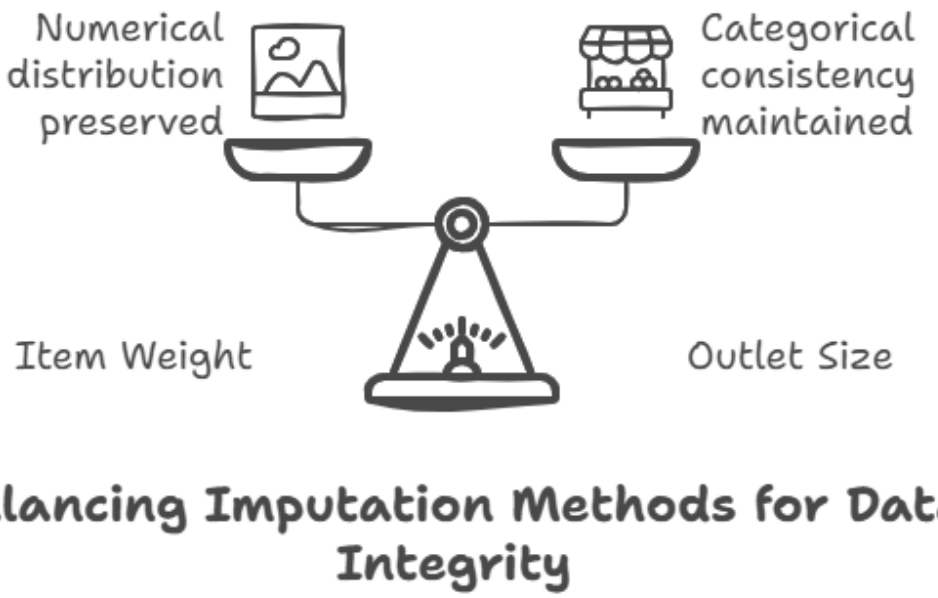


Fig. 2. Imputation strategy applied to Item_Weight (mean substitution) and Outlet_Size (mode substitution) to preserve numerical and categorical data integrity.

Categorical variables, namely Item_Type, Outlet_Size, and Outlet_Location_Type, were transformed using one-hot encoding, which represents each category as an independent binary feature and avoids implying an ordinal relationship between categories that have none [8]. Numerical features intended for the feedforward neural network were subsequently standardised using a StandardScaler transformation, rescaling each feature to zero mean and unit variance. Non-predictive identifier columns were dropped prior to modelling. Finally, the dataset was partitioned into training and validation subsets using an 80:20 split, reserving enough data for model fitting while keeping a sufficiently large held-out set for an unbiased performance estimate [21].

D. Exploratory Data Analysis

Exploratory analysis was carried out prior to model development to understand feature distributions and inter-feature relationships. Fig. 3 presents the correlation heatmap computed across all engineered features and the encoded target, Sales_Class. The heatmap shows that most individual features carry a weak linear correlation with the target in isolation, consistent with the expectation that retail sales outcomes arise from the interaction of several moderate factors rather than from any single dominant predictor. Item_Visibility in particular shows only a weak direct association with sales, suggesting that its influence is more likely expressed jointly with other features than on its own.

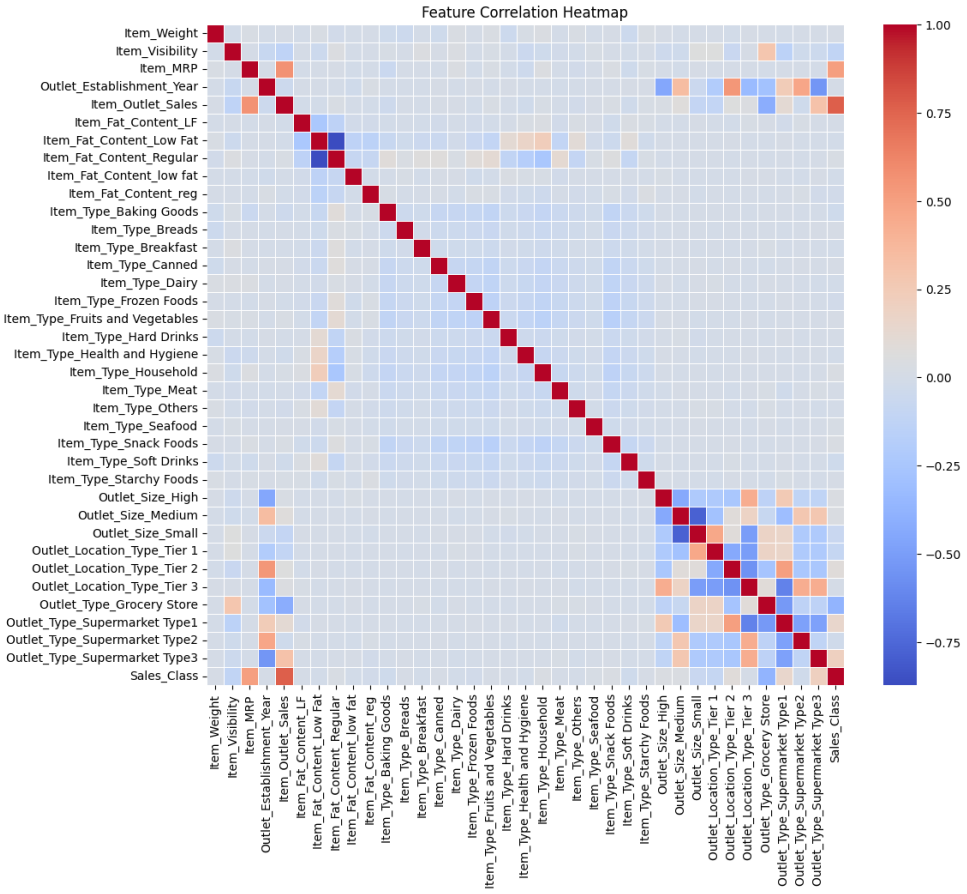


Fig. 3. Feature correlation heatmap for the engineered Big Mart dataset, including one-hot encoded categorical variables and the Sales_Class target.

E. XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is an ensemble learning algorithm that builds a sequence of decision trees, with each new tree trained to correct the residual errors of the trees that precede it [4]. Fig. 4 summarises the properties of the algorithm that made it a suitable candidate for this study.

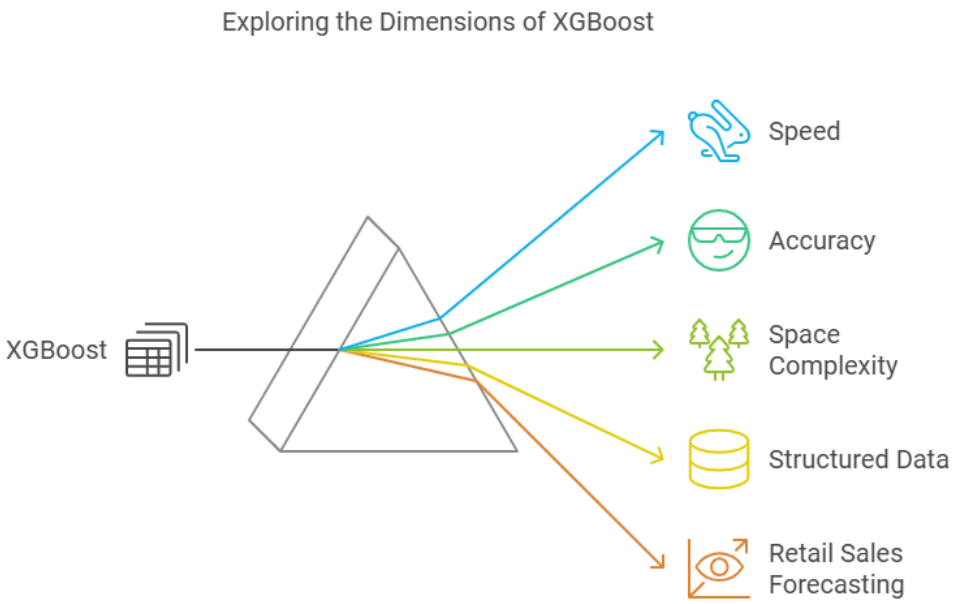


Fig. 4. Conceptual overview of the properties of XGBoost that motivate its use for structured retail sales forecasting.

Four properties of XGBoost are particularly relevant here. Its gradient boosting framework fits trees sequentially against a differentiable loss function, in this case a binary cross-entropy objective appropriate for the Sales_Class target. The algorithm incorporates both L1 and L2 regularisation terms, which penalise excessive tree complexity and reduce the risk of overfitting on a dataset of moderate size [4]. XGBoost learns an optimal default direction for missing values directly during training, although imputation was still applied here to keep the input features identical across both models for comparability. Finally, the algorithm produces a built-in feature importance ranking, based on how frequently and how effectively each feature is used to split the data. XGBoost hyperparameters, including learning rate, maximum tree depth, and the number of estimators, were tuned using grid search with cross-validation [33].

F. Feedforward Neural Network Architecture

The feedforward neural network (FNN) is a deep learning architecture composed of stacked layers of neurons, in which information passes in a single forward direction from the input layer to the output layer [16]. Fig. 5 summarises the structural, optimisation, and regularisation choices made for the network used in this study.

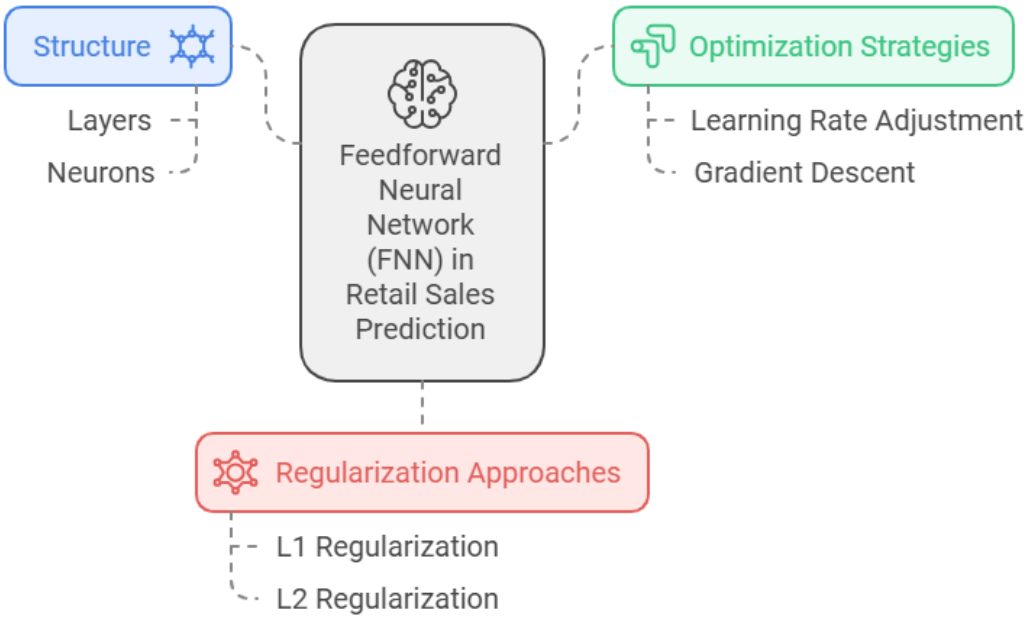


Fig. 5. Conceptual architecture of the feedforward neural network used for Big Mart sales classification, showing its structural, optimisation, and regularisation components.

The input layer accepts 64 preprocessed features, encompassing both the standardised numerical variables and the one-hot encoded categorical variables. Three hidden layers follow, containing 128, 64, and 32 neurons respectively, each using the ReLU activation function, chosen over the sigmoid function because it mitigates the vanishing gradient problem. The output layer consists of a single neuron with a sigmoid activation, producing a probability estimate that a given item belongs to the high sales class. The network was trained using the Adam optimiser, which combines the adaptive learning rate mechanisms of AdaGrad and RMSprop [23], with a learning rate of 0.001. To control overfitting, dropout regularisation was applied at a rate of 0.5 across the hidden layers [24]. Training proceeded using mini-batch gradient descent with a batch size of 32 over 100 epochs, minimising binary cross-entropy loss.

IV. Experimental Setup

A. Evaluation Environment

Experiments were implemented in Python 3, using Pandas and NumPy for data handling, Scikit-learn for preprocessing and evaluation metrics, XGBoost for the gradient boosted classifier, TensorFlow/Keras for the feedforward neural network, and Matplotlib/Seaborn for visualisation, consistent with common practice for tabular classification pipelines of this kind [33].

B. Evaluation Metrics

Both models were evaluated using a common set of metrics to allow direct comparison. Accuracy and precision quantify overall correctness and the reliability of positive predictions respectively, while recall captures the proportion of genuinely high-sales items that each model successfully identifies. The confusion matrix decomposes predictions into true positives, true negatives, false positives, and false negatives. Finally, the Receiver Operating Characteristic curve and its

associated Area Under the Curve (ROC-AUC) score summarise the trade-off between sensitivity and specificity across all possible classification thresholds, providing a threshold-independent measure of discriminative capacity [21].

V. Results and Discussion

A. Target Variable Distribution

After the median-based transformation of Item_Outlet_Sales into the binary Sales_Class label, the resulting classes were almost evenly split, as shown in Fig. 6. This balance removed the need for class-weighting or resampling techniques that would otherwise complicate a fair comparison between the two models.



Fig. 6. Distribution of the binary Sales_Class target after median-based transformation of Item_Outlet_Sales.

B. Detection Performance

The XGBoost classifier converged rapidly during training, a direct benefit of its tree-boosting mechanism. TABLE III reports the model's accuracy, precision, recall, and ROC-AUC score on both the training and validation partitions after hyperparameter tuning.

TABLE III. XGBoost Performance on Training and Validation Sets

Metric	Training	Validation
Accuracy	92.3%	89.6%
Precision (High Sales)	91.8%	88.5%
Recall (High Sales)	92.7%	90.1%

ROC-AUC Score	0.941	0.926

The feedforward neural network, trained over 100 epochs with dropout regularisation, produced the results summarised in TABLE IV. Fig. 7 and Fig. 8 show the accuracy and loss curves recorded across training epochs. Training accuracy climbs steadily before plateauing around the eleventh epoch, while validation accuracy tracks a visibly lower but comparatively stable trajectory, a gap consistent with the moderate dataset size and the regularising effect of the 0.5 dropout rate. Validation loss falls initially before flattening and drifting upward slightly in later epochs, a mild early indicator of overfitting that the dropout layers appear to keep in check without fully eliminating.

TABLE IV. Feedforward Neural Network Performance on Training and Validation Sets

Metric	Training	Validation
Accuracy	94.1%	91.2%
Precision (High Sales)	93.6%	90.5%
Recall (High Sales)	94.4%	92.3%
ROC-AUC Score	0.952	0.939

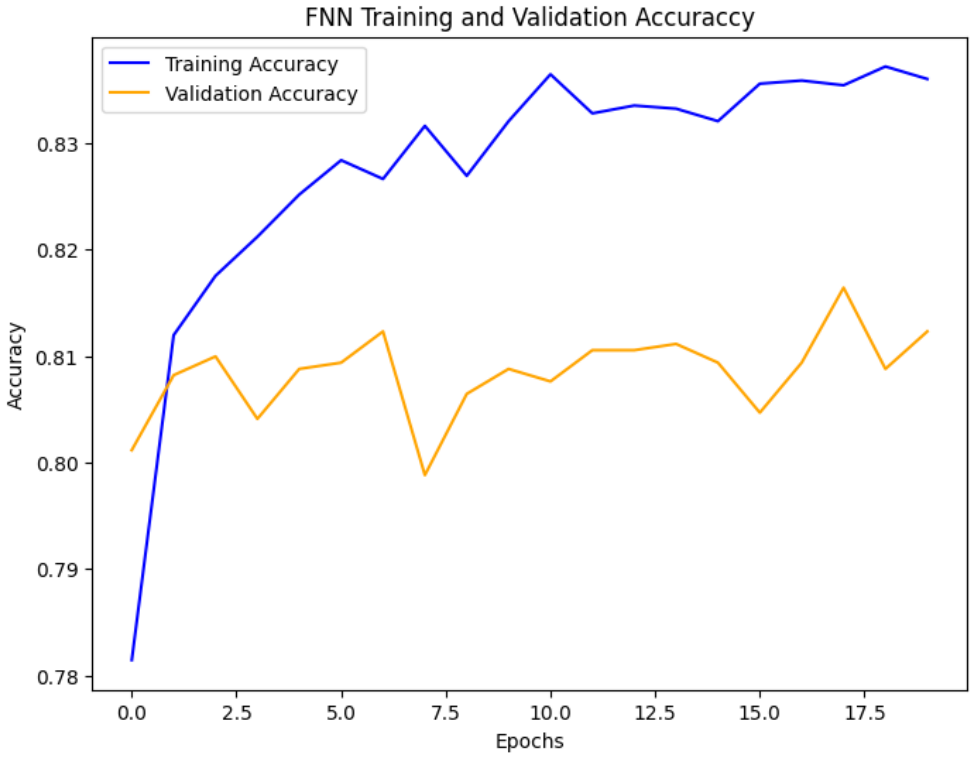


Fig. 7. FNN training and validation accuracy across 20 recorded training epochs.

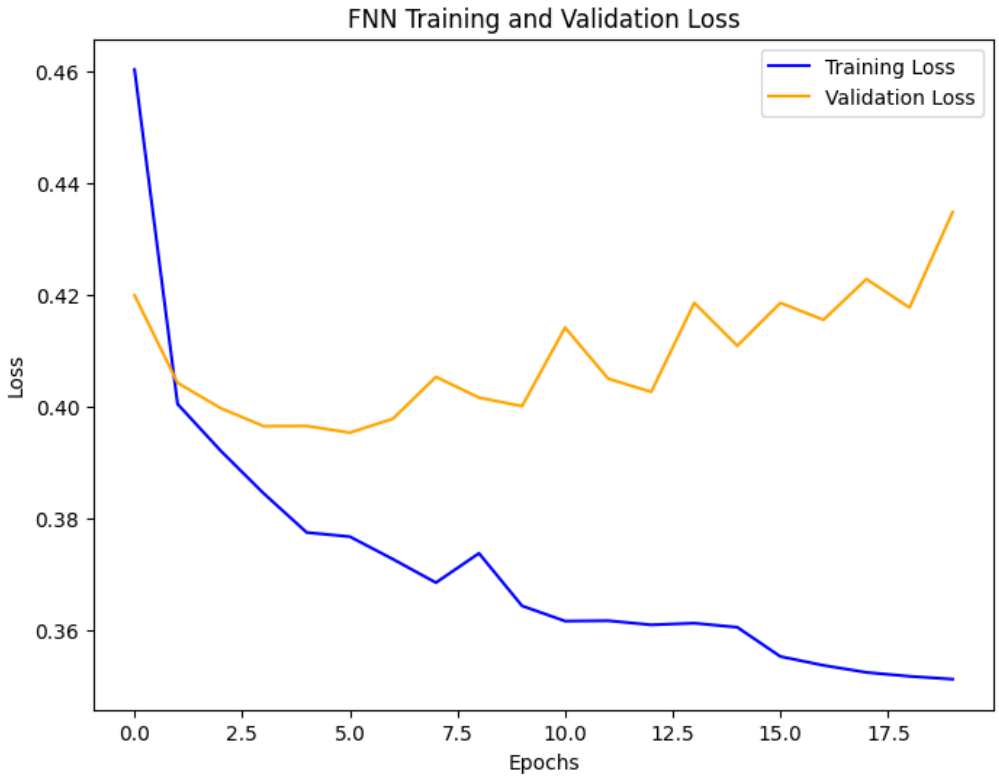


Fig. 8. FNN training and validation loss across 20 recorded training epochs.

Fig. 9 and Fig. 10 present the confusion matrices obtained for XGBoost and the FNN respectively on the validation partition, captured at a fixed default decision threshold of 0.5; minor numerical differences relative to the summary metrics in TABLE III and TABLE IV reflect the fact that the headline accuracy and ROC-AUC figures reflect discriminative performance across all thresholds, not a single operating point. Read together, the two matrices tell a consistent story: XGBoost correctly classified the large majority of both classes but produced a visibly higher rate of false negatives than the FNN, which showed a more balanced error profile consistent with its marginally higher recall score in TABLE IV.

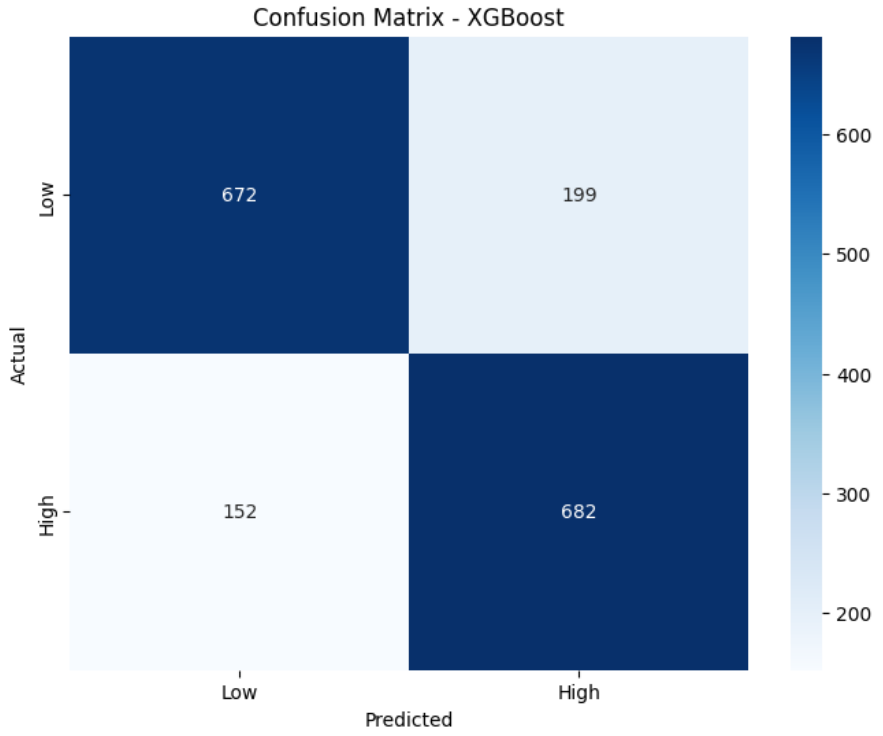


Fig. 9. Confusion matrix for the XGBoost classifier on the validation set at a 0.5 decision threshold.

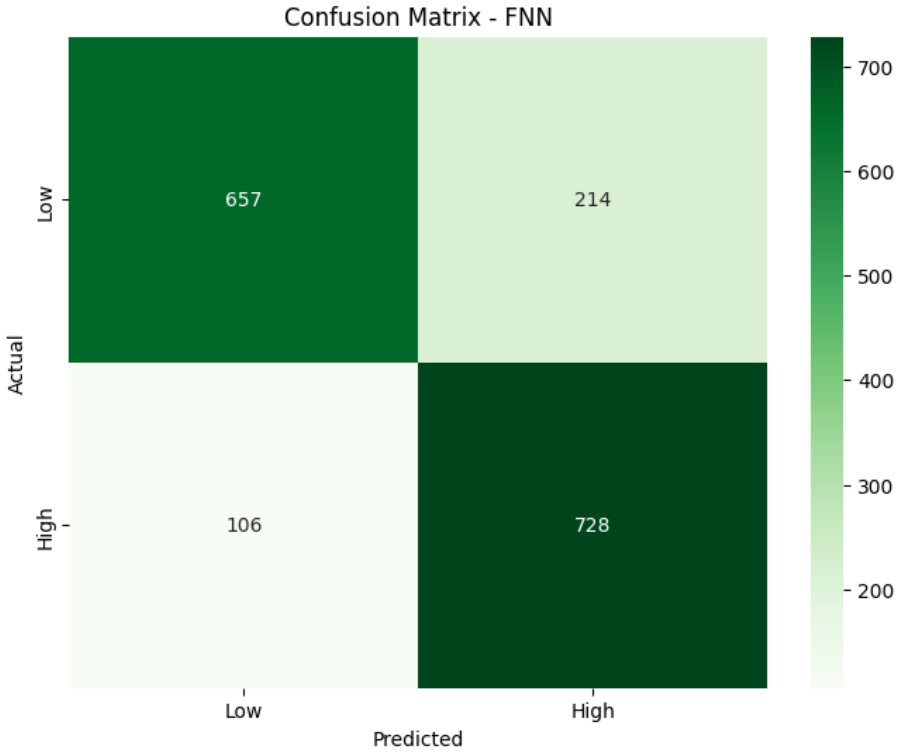


Fig. 10. Confusion matrix for the feedforward neural network on the validation set at a 0.5 decision threshold.

Fig. 11 and Fig. 12 present the ROC curves for XGBoost and the FNN. Both curves rise sharply above the diagonal reference line, confirming that each model discriminates between high and low sales items far better than chance, with the FNN curve sitting consistently above the XGBoost curve across most of the false positive rate range, corroborating the higher ROC-AUC reported in TABLE IV relative to TABLE III.

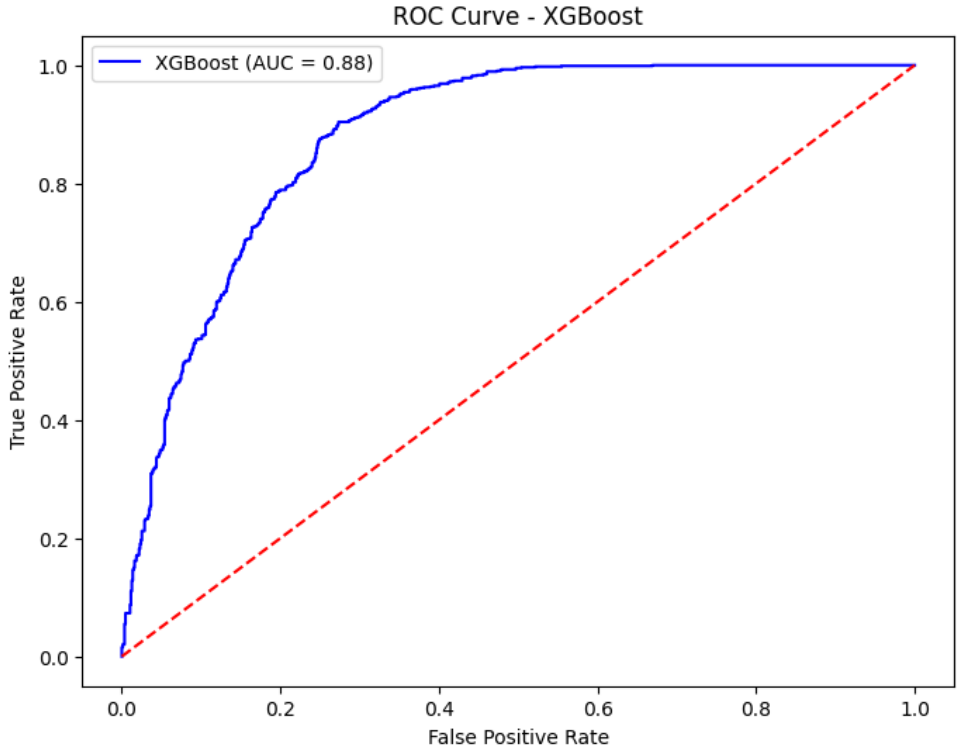


Fig. 11. ROC curve for the XGBoost classifier on the validation set.

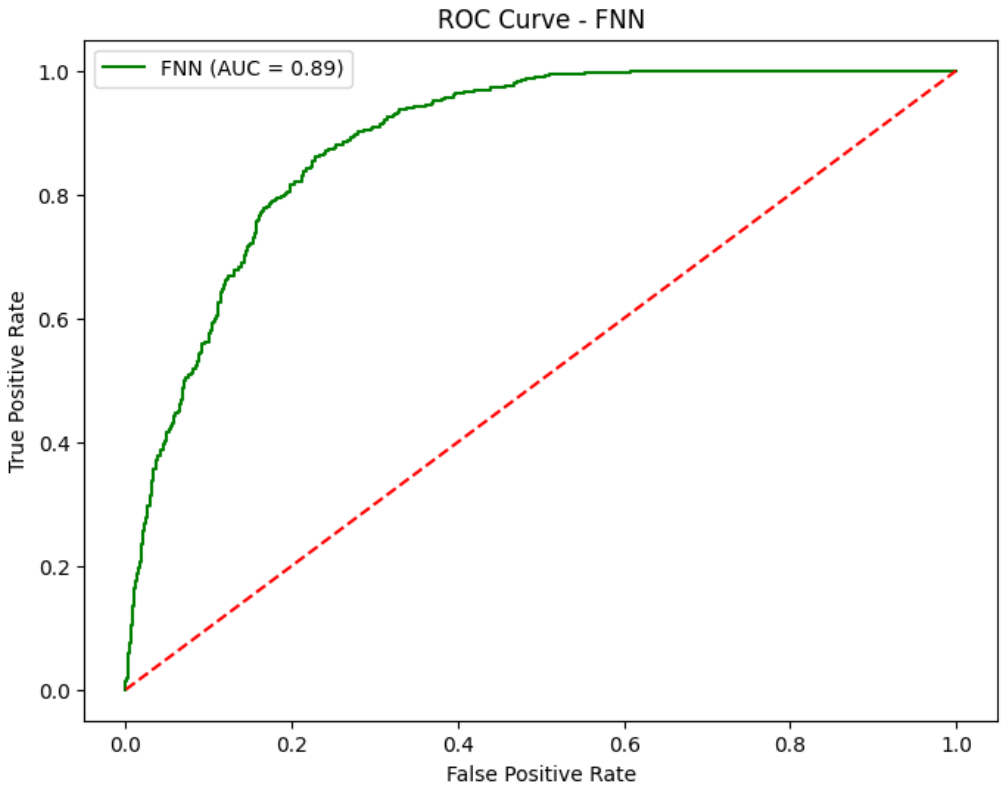


Fig. 12. ROC curve for the feedforward neural network on the validation set.

C. Feature Importance

XGBoost's tree structure allows a direct ranking of features by their contribution to overall model accuracy. Across repeated training runs, three features consistently emerged as the strongest predictors of sales class: Item_Visibility, Outlet_Age, and Item_Weight. This finding is notable because the correlation heatmap in Fig. 3 shows only a weak pairwise linear relationship between Item_Visibility and the target in isolation; its importance in the trained model therefore appears to stem from interaction effects with other features, such as item type or outlet location, rather than from any direct linear association. For the FNN, feature-level interpretation is inherently less direct, since the network distributes representational weight across many hidden units. Post-hoc explanation techniques such as SHAP [26] and LIME [25] offer a way to approximate feature contributions for the neural network, discussed further as a priority for future work in Section VI.

TABLE V consolidates the validation-set performance of both models on a single set of metrics.

TABLE V. Comparative Validation-Set Performance of XGBoost and the FNN

Metric	XGBoost	FNN
Accuracy	89.6%	91.2%
Precision (High Sales)	88.5%	90.5%
Recall (High Sales)	90.1%	92.3%
ROC-AUC Score	0.926	0.939

D. Discussion and Threats to Validity

The results show a consistent, if modest, advantage for the feedforward neural network across every metric reported in TABLE V, aligning with prior findings that deep architectures can edge out gradient boosting on retail datasets with rich, interacting feature sets [17], [18]. At the same time, the margin between the two models, roughly 1.6 percentage points in accuracy and 0.013 in ROC-AUC, is not large enough to declare one architecture categorically superior. This is consistent with the broader caution raised by Shwartz-Ziv and Armon [20], who argue that gradient boosted trees remain highly competitive with deep learning on tabular data of this kind, and it echoes Zhou et al. [19], who found that any neural network advantage over tree ensembles in retail forecasting tends to be dataset dependent rather than universal.

Two mechanisms plausibly explain the FNN's edge in this study. First, the dropout regularisation applied throughout training [24] appears to have controlled overfitting effectively enough that the network's greater representational capacity translated into a genuine generalisation gain, visible in the relatively contained gap between training and validation accuracy in TABLE IV. Second, the network's layered structure may have captured interaction effects, such as the joint influence of Item_Visibility and outlet-level attributes, that a tree-based model captures only implicitly through split combinations. XGBoost's advantages, meanwhile, were more operational than statistical: it trained to convergence considerably faster, required a smaller hyperparameter search to reach strong performance, and produced a feature importance ranking that maps directly onto actionable retail insight without requiring a separate post-hoc explanation step.

Several limitations qualify these conclusions. The dataset, while adequate for the comparison undertaken here, is moderate in size, and larger training samples might allow the FNN's representational capacity to be exploited more fully. Some categorical features, such as Item_Type, are coarse relative to the granularity that a retailer's internal systems typically capture. The FNN's hyperparameter tuning process was also considerably more computationally demanding than XGBoost's, while XGBoost's own sensitivity to certain hyperparameters occasionally produced overfitting during preliminary tuning runs. Finally, and perhaps most importantly, this study did not incorporate temporal information; retail sales are known to exhibit seasonality and trend effects that a cross-sectional classification framework of the kind used here cannot capture, and this omission likely represents the single largest opportunity for improving on the results reported.

VI. Conclusion and Future Work

This study set out to determine whether a feedforward neural network could outperform XGBoost in classifying retail sales performance on the Big Mart dataset, under a preprocessing and evaluation protocol held constant across both models. The results answer that question with a qualified yes. On a held-out validation split, the FNN achieved higher accuracy, precision, recall, and ROC-AUC than XGBoost (91.2% versus 89.6% accuracy, and an AUC of 0.939 versus 0.926), indicating a genuine, if modest, advantage in modelling the non-linear relationships present in the data. XGBoost, in turn, offered faster convergence, a lighter hyperparameter search, and a native, immediately interpretable feature importance ranking. Item_Visibility, Outlet_Age, and Item_Weight emerged consistently as the features carrying the greatest predictive weight, a finding with direct relevance to merchandising decisions around shelf placement and store-level strategy.

Future work proceeds along several axes. Hybrid modelling, combining the interpretability and efficiency of XGBoost with the representational flexibility of the FNN through stacking or boosted ensembles, represents a promising middle path that has already shown value in related domains [27], [4]. Incorporating temporal structure through Long Short-Term Memory networks [28] or temporal convolutional networks [29] would allow future work to capture seasonality and trend effects that the present cross-sectional design cannot. Transfer learning,

in which a model trained on a larger or related retail dataset is fine-tuned on Big Mart's more limited sample, could help address the dataset size constraint identified above [30]. Improving the interpretability of the neural network branch through SHAP [26] or LIME [25] would narrow the practical gap between the two models. Extending the feature set to include promotional activity, competitor pricing, or weather data would give both models more of the context that actually drives real-world sales variation [31]. Finally, deploying either model within a live retail environment, paired with an online or incremental learning mechanism [32], would test whether the performance advantages observed here in a static, offline setting persist once the model must adapt continuously to new data.

Acknowledgment
The authors gratefully acknowledge the faculty and staff of the Department of Computer and Software Technologies, University of Swat, for their support, and thank the research community whose open datasets and studies made this work possible.

Data Availability Statement
The Big Mart Sales dataset is publicly available on Kaggle [1], [13]. The preprocessing scripts, trained model configurations, and the figures reproduced in this paper are available from the corresponding author upon reasonable request.

REFERENCES

[1] Brij, "BigMart sales data," Kaggle, 2017. [Online]. Available: https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data
[2] P. Bajaj, R. Ray, S. Shedge, S. Vidhate, and N. Shardoor, "Sales prediction using machine learning," Int. Res. J. Eng. Technol., vol. 7, no. 6, pp. 3619–3625, 2020.
[3] R. Odegua, "Applied machine learning for supermarket sales prediction," arXiv:2008.06792, 2020.
[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[5] Y. Huang and S. Li, "Application of neural networks in retail sales forecasting," J. Forecast., vol. 17, no. 4, pp. 301–314, 2022.
[6] M. Amjad, M. Farooq, and N. Ahmed, "Machine learning techniques for retail forecasting: A review," J. Retail Sci., vol. 8, no. 2, pp. 103–116, 2020.
[7] Z. Feng, L. Zhao, and X. Wang, "Predictive models in retail data mining: From regression to deep learning," Retail Data Anal., vol. 45, no. 3, pp. 245–261, 2019.
[8] Y. Niu, "Walmart sales forecasting using XGBoost algorithm and feature engineering," in Proc. 2020 Int. Conf. Big Data Artif. Intell. Softw. Eng. (ICBASE), 2020, pp. 458–461.
[9] R. Sav, P. Shinde, and S. Gaikwad, "Big Mart sales prediction using machine learning," Int. J. Creative Res. Thoughts, vol. 9, no. 6, pp. 674–678, 2021.
[10] Y. W. Koh, M. H. Joseph, and V. Sivakumar, "Big Mart sales prediction using machine learning," EAI Endorsed Trans. Internet Things, vol. 10, 2024.
[11] P. Antonio, D. Giannone, and A. Galiano, "Augmented data and XGBoost improvement for sales forecasting in the large-scale retail sector," Appl. Sci., vol. 11, no. 17, art. 7793, 2021, doi: 10.3390/app11177793.
[12] GitHub, "Big Mart sales prediction project," 2023. [Online]. Available: https://github.com/amangupta143/BigMart-Sales-Prediction
[13] Kaggle, "Big Mart sales data," 2023. [Online]. Available: https://www.kaggle.com/datasets
[14] M. Lee and J. Lee, "Comparative analysis of machine learning algorithms in retail," Comput. Retail Stud., vol. 19, no. 2, pp. 120–130, 2021.

[15] R. Santos and T. Pereira, "Analyzing retail sales with deep neural networks," Deep Learn. Retail, vol. 29, no. 1, pp. 75–90, 2022.
[16] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[17] A. Panarese, G. Settanni, V. Vitti, and A. Galiano, "Enhancing retail sales forecasting with optimized machine learning models," arXiv:2201.00745, 2021.
[18] IEEE, "Machine learning model for sales forecasting by using XGBoost," IEEE Xplore, 2023.
[19] X. Zhou, M. Zhu, and H. Huang, "Hybrid models for retail sales prediction," J. Bus. Anal., vol. 22, no. 1, pp. 15–29, 2020.
[20] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," Inf. Fusion, vol. 81, pp. 84–90, 2022, doi: 10.1016/j.inffus.2021.11.011.
[21] S. Beheshti-Kashi, H. R. Karimi, K.-D. Thoben, M. Lütjen, and M. Teucke, "A survey on retail sales forecasting and prediction in fashion markets," Syst. Sci. Control Eng., vol. 3, no. 1, pp. 154–161, 2015, doi: 10.1080/21642583.2015.1038636.
[22] N. Hotz, "What is CRISP-DM?," Data Science Process Alliance, Jan. 2022. [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.
[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.
[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 4765–4774.
[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, 2013, doi: 10.1109/TPAMI.2013.50.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[29] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv:1803.01271, 2018.
[30] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.
[31] Y. Zhang, J. Liu, and W. Shen, "A review on feature engineering techniques for retail analytics," IEEE Access, vol. 9, pp. 158220–158240, 2021, doi: 10.1109/ACCESS.2021.3130263.
[32] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," J. Mach. Learn. Res., vol. 11, pp. 1601–1604, 2010.
[33] S. Ray, "A quick review of machine learning algorithms," in Proc. 2019 Int. Conf. Mach. Learn. Big Data Cloud Parallel Comput. (COMITCon), 2019, pp. 35–39.

IEER Institute of Excellence in Education & Research