Sales Analysis and Forecasting over Time Using Machine Learning Models

Trần Đại Hải[1][0000-0000-2352-0420], Trần Tuyết Như[1][0000-0000-02352-1129], Lê Phúc Thịnh[1][0000-0000-2352-1499], Mai Nguyễn Bảo Duy[1][0000-0000-2352-0369], and Lê Nguyễn Thành Công[1][0000-0000-2352-0191]

University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam

23520420@gm.uit.edu.vn, 23521129@gm.uit.edu.vn, 23521499@gm.uit.edu.vn, 23520369@gm.uit.edu.vn, 23520191@gm.uit.edu.vn

Abstract. Currently, the retail industry is facing rapid changes in consumer behavior and global market trends. Forecasting sales performance plays a crucial role in supporting data-driven decision-making for businesses. In this paper, we conduct a comprehensive sales analysis and forecasting study based on the *Global Fashion Retail Stores Dataset* from Kaggle, which simulates the operation of a multinational fashion retail chain from January 2023 to March 2025 across seven countries. The dataset includes transaction-level information such as time, store location, customer demographics, product details, and sales revenue.

We apply various machine learning and statistical models — including LightGBM, XGBoost, Random Forest, SARIMAX, and deep learning models like N-HiTS and N-BEATS — to predict future sales performance. Evaluation metrics such as MAE, RMSE, and sMAPE are used to assess model accuracy. Experimental results are expected to show that ensemble learning models (e.g., LightGBM, XGBoost) and deep neural forecasting approaches (e.g., N-HiTS) outperform traditional models in short-term and medium-term predictions. This study aims to identify key factors influencing retail performance and to propose data-driven strategies that help businesses enhance operational efficiency and profitability.

Keywords: Sales forecasting · Machine learning · Fashion retail analytics · Time series prediction · LightGBM · XGBoost · Random Forest · N-HiTS · N-BEATS · SARIMAX

1 Introduction

The retail industry plays a vital role in the global economy, contributing significantly to employment, consumer spending, and business growth. With the increasing availability of large-scale transactional data, sales forecasting has become an essential analytical tool for improving business planning and inventory management. Accurately predicting sales helps retailers anticipate market demand, optimize stock levels, and design effective marketing strategies.

However, retail sales forecasting remains a challenging problem due to the presence of seasonality, promotions, regional differences, and unpredictable consumer behavior. Traditional statistical methods such as SARIMA can capture linear temporal patterns but often fail to model complex and nonlinear relationships in large and multidimensional datasets.

Recent advances in machine learning and deep learning techniques — such as ensemble methods (LightGBM, XGBoost, Random Forest) and neural forecasting models (N-HiTS, N-BEATS) — provide new opportunities for improving sales prediction accuracy.

In this research, we focus on the *Global Fashion Retail Stores Dataset*, a synthetic dataset representing sales transactions across 35 stores in seven countries. Our goal is to analyze the dataset, explore the factors influencing sales performance, and compare multiple forecasting models to determine the most effective approach for predicting short-term and long-term sales trends. The findings from this study can provide practical insights for data-driven decision-making in the fashion retail industry.

2 Related Work

Retail sales forecasting has been extensively studied using various statistical and machine learning approaches. Previous research has demonstrated the effectiveness of different models across various retail domains and forecasting horizons.

2.1 Traditional Statistical Models

Traditional time series models have long been employed in retail forecasting. **Fildes et al. (2019)** conducted a comprehensive review of retail demand forecasting, examining challenges at different organizational levels and finding that while complex methods have been proposed, simpler approaches often remain competitive in practical applications.

Arunraj & Ahrens (2015) developed a hybrid SARIMA and Quantile Regression model for daily food sales forecasting, demonstrating that combining statistical time series models with distributional forecasting can provide valuable insights for retail applications with high volatility.

2.2 Machine Learning Approaches

Machine learning methods have shown remarkable success in retail forecasting. **Ganguly & Mukherjee (2024)** systematically compared various ML regression models, finding that optimized Random Forest achieved significantly higher accuracy compared to linear baselines, underscoring the capability of ensemble tree methods to capture complex patterns in retail data.

The M5 Competition **(Makridakis et al., 2022)** provided compelling evidence for the dominance of machine learning methods in large-scale retail forecasting, with gradient boosting approaches like LightGBM achieving approximately 14-20% lower error than statistical benchmarks through extensive feature engineering and cross-learning.

2.3 Deep Learning Approaches

Deep learning methods have emerged as powerful tools for retail sales forecasting. **Eglite & Birzniece (2022)** conducted a systematic review of deep learning applications in retail forecasting, cataloging various architectures including RNNs, CNNs, and hybrid models while discussing trade-offs between accuracy and interpretability.

Mansur et al. (2025) developed a CNN-LSTM hybrid network that incorporated external variables including holidays and weather conditions, achieving a MAPE of 4.16% and demonstrating the value of contextual information in retail forecasting.

2.4 Research Gap

While substantial research exists on retail forecasting, comprehensive comparisons of diverse models specifically for fashion retail with its unique seasonal patterns, demographic factors, and fast-changing trends remain limited. Our study addresses this gap by systematic evaluation of multiple forecasting paradigms on a comprehensive global fashion retail dataset.

3 Data Overview

3.1 Data Source and Description

This study utilizes the *Global Fashion Retail Sales* dataset, a comprehensive synthetic dataset sourced from Kaggle that simulates the operational dynamics of a multinational fashion retail chain. The dataset is specifically designed to support analytical tasks in retail performance analysis, customer behavior modeling, and sales forecasting.

The primary data file, `transactions.csv`, contains detailed transactional records spanning from **January 1, 2023, to March 18, 2025**, covering approximately 27 months of retail operations. This temporal range encompasses multiple business cycles, seasonal patterns, and promotional periods, providing a robust foundation for time series analysis and forecasting.

3.2 Data Structure and Attributes

The dataset is structured around a central transactions table with the following key attributes:

Table 1: Data Structure of Transactions Table

Variable Name	Data Type	Description
Invoice ID	String	Unique identifier for each transaction invoice
Customer ID	String	Anonymous identifier for customer tracking
Date	Date	Transaction date (YYYY-MM-DD)
Store ID	String	Identifier for the physical store location
Employee ID	String	Identifier for the attending staff member
Currency	String	Transaction currency (USD, CNY, GBP, EUR)
Transaction Type	String	Type of transaction (e.g., Sale, Return)
Product ID	String	Unique identifier for purchased items
Line Total	Float	Total monetary value for the product line

The dataset exhibits a multi-currency structure, requiring standardization for consolidated analysis. Exchange rates were applied as follows: CNY to USD (0.14), GBP to USD (1.33), and EUR to USD (1.16), with USD transactions remaining at parity (1.00).

3.3 Data Scale and Scope

The dataset represents substantial retail operations with the following dimensions:

- **Transaction Volume**: The dataset contains 808 unique daily aggregated sales records after processing, derived from individual transaction-level data.
- **Geographical Coverage**: Operations span 7 countries, providing diverse market perspectives and enabling cross-regional analysis.
- **Temporal Coverage**: 808 days of continuous operation from 2023-01-01 to 2025-03-18, capturing multiple seasonal cycles and business trends.
- **Multi-currency Operations**: Transactions occur in 4 different currencies, reflecting the global nature of the retail operations.

3.4 Data Preprocessing and Aggregation

To prepare the data for time series forecasting, we implemented a comprehensive preprocessing pipeline:

Currency Standardization All transactional values were converted to US Dollars (USD) using fixed exchange rates to ensure consistency in monetary analysis. Missing exchange rates were handled by defaulting to USD parity (1.0).

Temporal Aggregation Individual transactions were aggregated to daily sales totals, creating a unified time series for forecasting. The aggregation was performed as follows:

$$\text{daily\_total\_usd} = \sum_{i=1}^{n} (\text{Line Total}_i \times \text{exchange\_rate}_i) \tag{1}$$

where n represents the number of transactions per day, and exchange_rate_i converts each transaction to USD.

Data Quality Assurance The preprocessing phase included handling of missing values through forward-filling techniques and validation of temporal consistency. The resulting daily sales time series forms the foundation for subsequent feature engineering and model development.

3.5 Exploratory Data Analysis

Preliminary analysis of the aggregated daily sales data reveals several key characteristics of the retail operations:

- **Temporal Patterns**: The data exhibits clear weekly seasonality with peak sales typically occurring on weekends, consistent with fashion retail consumer behavior.
- **Seasonal Trends**: Pronounced quarterly patterns align with traditional retail calendar events, including holiday seasons and promotional periods.
- **Operational Scale**: Daily sales volumes demonstrate the substantial scale of operations, with fluctuations reflecting both seasonal demand and potential market dynamics.

The processed dataset, comprising 808 daily observations with standardized USD values, provides a robust foundation for developing and evaluating forecasting models across multiple temporal horizons and geographical segments.

4 Methodology

This section presents the comprehensive methodology employed for sales forecasting in the global fashion retail domain. We describe the theoretical foundations of all models implemented, their relevance to retail forecasting, and the experimental setup for comparative evaluation.

4.1 Forecasting Models

Seasonal ARIMA with Exogenous Variables (SARIMAX) The SARIMAX model extends the classical ARIMA framework by incorporating both seasonal patterns and exogenous variables. For a time series $\{y_t\}$ with exogenous variables $\{X_t\}$, the SARIMAX$(p, d, q)(P, D, Q)_s$ model is defined as:

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t + \beta X_t \tag{2}$$

where:

- $\phi_p(B)$ and $\theta_q(B)$ are the non-seasonal AR and MA polynomials
- $\Phi_P(B^s)$ and $\Theta_Q(B^s)$ are the seasonal AR and MA polynomials
- $(1-B)^d$ and $(1-B^s)^D$ are the non-seasonal and seasonal difference operators
- s is the seasonal period (e.g., 7 for weekly seasonality)
- β represents the coefficients for exogenous variables X_t

The model components are defined as follows:
Autoregressive (AR) Component:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \tag{3}$$

Moving Average (MA) Component:

$$y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{4}$$

Differencing (I) Component:

$$\nabla y_t = y_t - y_{t-1} \tag{5}$$

The complete SARIMAX$(p, d, q)(P, D, Q)_s$ model combines these elements:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t - \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \beta X_t \tag{6}$$

For retail sales forecasting, we incorporate exogenous variables including promotional flags, holiday indicators, day-of-week effects, and store-specific characteristics. The model selection follows the Box-Jenkins methodology with automatic parameter selection using the Akaike Information Criterion (AIC).

Random Forest Random Forest (RF) is an ensemble learning method that constructs multiple decision trees during training and outputs the mean prediction of individual trees for regression tasks. Given a training set $\{(x_i, y_i)\}_{i=1}^{n}$, the RF prediction is:

$$\hat{y}_{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{7}$$

where $T_b(x)$ is the prediction of the b-th tree and B is the number of trees. Each tree is trained on a bootstrap sample of the training data, with feature randomization at each split to de-correlate trees.

The splitting criterion at each node minimizes the impurity measure. For regression tasks, we use Mean Squared Error (MSE):

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2 \tag{8}$$

where Q_m represents the data at node m, n_m is the number of samples, and $\bar{y}_m$ is the mean value.

Key advantages of Random Forest for retail forecasting include:

- Handling of non-linear relationships and complex interactions
- Robustness to outliers and missing values
- Natural feature importance assessment
- No requirement for extensive data preprocessing

Extreme Gradient Boosting (XGBoost) XGBoost is an optimized gradient boosting implementation that minimizes a regularized objective function. The model at iteration t is:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{9}$$

where l is a differentiable convex loss function and $\Omega(f_t)$ is the regularization term:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{10}$$

The second-order Taylor expansion enables efficient optimization:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{11}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the first and second derivatives of the loss function, respectively.

The algorithm proceeds as follows:

1. Initialize model with a constant value: $\hat{y}_i^{(0)} = \arg\min_\gamma \sum_{i=1}^{n} l(y_i, \gamma)$
2. For $t = 1$ to T:
   (a) Compute gradients g_i and hessians h_i for $i = 1, \ldots, n$
   (b) Fit a weak learner $f_t(x)$ to the targets $-g_i/h_i$
   (c) Compute optimal weight w_t^* for the new tree
   (d) Update the model: $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta w_t^* f_t(x_i)$

Light Gradient Boosting Machine (LightGBM) LightGBM improves upon traditional gradient boosting through two key techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). The objective function follows:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{12}$$

Gradient-based One-Side Sampling (GOSS) keeps all instances with large gradients and randomly samples instances with small gradients:

$$A = \{i \mid |g_i| \geq g_\epsilon\}, \quad B = \text{RandomSample}(\{i \mid |g_i| < g_\epsilon\}, r \cdot n) \tag{13}$$

Exclusive Feature Bundling (EFB) bundles mutually exclusive features to reduce dimensionality:

$$\mathcal{F} = \{f_1, f_2, \ldots, f_m\} \to \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_k\} \tag{14}$$

LightGBM employs leaf-wise tree growth, selecting the leaf with maximum delta loss for splitting:

$$\mathcal{G} = \frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \tag{15}$$

N-BEATS The Neural Basis Expansion Analysis for Time Series (N-BEATS) is a deep learning architecture based solely on fully connected layers. The model operates through multiple stacks of blocks, each producing two outputs: a backcast $\hat{\mathbf{x}}_\ell$ and a forecast $\hat{\mathbf{y}}_\ell$.

Each block ℓ with input $\mathbf{x}_\ell$ performs:

$$\mathbf{h}_{\ell,1} = \text{FC}_{\ell,1}(\mathbf{x}_\ell), \quad \mathbf{h}_{\ell,2} = \text{FC}_{\ell,2}(\mathbf{h}_{\ell,1}) \tag{16}$$

$$\mathbf{h}_{\ell,3} = \text{FC}_{\ell,3}(\mathbf{h}_{\ell,2}), \quad \mathbf{h}_{\ell,4} = \text{FC}_{\ell,4}(\mathbf{h}_{\ell,3}) \tag{17}$$

$$\theta_\ell^b = \text{LINEAR}_\ell^b(\mathbf{h}_{\ell,4}), \quad \theta_\ell^f = \text{LINEAR}_\ell^f(\mathbf{h}_{\ell,4}) \tag{18}$$

The forecasts are generated through basis expansion:

$$\hat{\mathbf{y}}_\ell = \sum_{i=1}^{\dim(\theta_\ell^f)} \theta_{\ell,i}^f \mathbf{v}_i^f, \quad \hat{\mathbf{x}}_\ell = \sum_{i=1}^{\dim(\theta_\ell^b)} \theta_{\ell,i}^b \mathbf{v}_i^b \tag{19}$$

The doubly residual stacking architecture ensures:

$$\mathbf{x}_\ell = \mathbf{x}_{\ell-1} - \hat{\mathbf{x}}_{\ell-1}, \quad \hat{\mathbf{y}} = \sum_\ell \hat{\mathbf{y}}_\ell \tag{20}$$

We implement both generic and interpretable configurations, with the latter enforcing trend and seasonality decomposition through polynomial and Fourier basis functions.

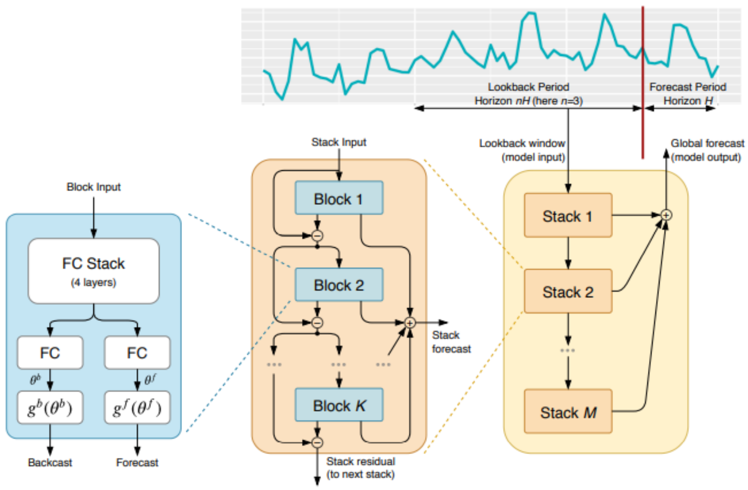

Fig. 1: Overview Architecture of N-BEATS

N-HiTS The Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS) extends N-BEATS with multi-rate sampling and hierarchical interpolation. Each block incorporates max pooling for multi-scale analysis:

$$y^{(p)}_{t-L:t,\ell} = \text{MaxPool}(y_{t-L:t,\ell}, k_\ell) \tag{21}$$

The model employs hierarchical interpolation to reduce forecast dimensionality:

$$\hat{y}_{\tau,\ell} = g(\tau, \theta^f_\ell), \quad \forall \tau \in \{t+1, \ldots, t+H\} \tag{22}$$

$$\tilde{y}_{\tau,\ell} = g(\tau, \theta^b_\ell), \quad \forall \tau \in \{t-L, \ldots, t\} \tag{23}$$

where g is an interpolation function and r_ℓ controls the expressiveness ratio. The linear interpolator is defined as:

$$g(\tau, \theta) = \theta[t_1] + \left(\frac{\theta[t_2] - \theta[t_1]}{t_2 - t_1}\right)(\tau - t_1) \tag{24}$$

The hierarchical forecast is obtained by:

$$\hat{y}_{t+1:t+H} = \sum_{\ell=1}^{L} \hat{y}_{t+1:t+H,\ell} \tag{25}$$

$$y_{t-L:t,\ell+1} = y_{t-L:t,\ell} - \tilde{y}_{t-L:t,\ell} \tag{26}$$

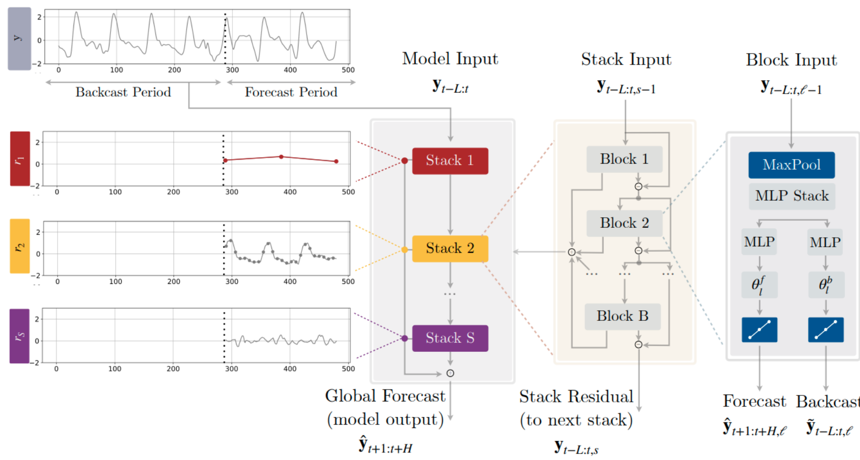

Fig. 2: Overview Architecture of N-HiTS

4.2 Experimental Setup

Data Preparation The Global Fashion Retail Stores Dataset spans from January 2023 to March 2025 across 35 stores in 7 countries. We aggregate transactions to daily sales at multiple hierarchical levels:

- Global level: Total daily revenue across all stores
- Country level: Daily revenue per country
- Store level: Daily revenue per individual store
- Product category level: Daily sales quantity per product category

All monetary values are converted to USD for consistency. We handle missing values through forward filling and remove outliers beyond three standard deviations using the Z-score method:

$$z = \frac{x - \mu}{\sigma}, \quad \text{remove if } |z| > 3 \tag{27}$$

Feature Engineering For machine learning models, we create comprehensive feature sets:

- **Temporal features**: day-of-week, month, quarter, year, is_weekend, is_holiday, day_of_year.
- **Lag features**: 1-day, 2-day, 7-day, 14-day, 30-day lags of target variable.
- **Rolling statistics**: 7-day and 30-day moving averages, standard deviations, minimum and maximum values.
- **Exponential weighted moving averages**: with halflife of 7 and 30 days.
- **Seasonal indicators**: seasonal dummy variables, Fourier terms for weekly and yearly seasonality.

We ensure temporal consistency by preventing data leakage through careful feature calculation. All lagged and rolling features are computed using only past information.

Training and Evaluation We employ temporal cross-validation with an expanding window approach to maintain temporal ordering:

- Training period: January 2023 - December 2024
- Validation period: January 2025 - February 2025
- Test period: March 2025

The expanding window validation scheme ensures that:

$$\text{Window}_k = \{\text{train: } [t_0, t_k], \text{ test: } [t_{k+1}, t_{k+H}]\} \tag{28}$$

Models are evaluated using multiple metrics to assess different aspects of forecast accuracy:

$$\text{MAE} = \frac{1}{H}\sum_{i=1}^{H}|y_{T+i} - \hat{y}_{T+i}| \tag{29}$$

$$\text{RMSE} = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(y_{T+i} - \hat{y}_{T+i})^2} \tag{30}$$

$$\text{MAPE} = \frac{100}{H}\sum_{i=1}^{H}\frac{|y_{T+i} - \hat{y}_{T+i}|}{|y_{T+i}|} \tag{31}$$

$$\tag{32}$$

Implementation Details All experiments are conducted using Python 3.9 with the following libraries and versions:

- scikit-learn 1.2.2 for traditional machine learning models
- statsmodels 0.13.5 for statistical models
- pmdarima 2.0.3 for automatic SARIMA model selection
- neuralforecast 1.5.0 for deep learning models
- xgboost 1.7.5 for XGBoost implementation
- lightgbm 3.3.5 for LightGBM implementation
- tensorflow 2.11.0 for LSTM and RNN implementations

Hyperparameter optimization is performed using Bayesian optimization with 50 iterations for each model. Key hyperparameters tuned include:

- **SARIMAX/SARIMA**: $(p, d, q)(P, D, Q)_s$ orders, seasonal period s
- **Random Forest**: `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`
- **XGBoost**: `learning_rate`, `max_depth`, `n_estimators`, `subsample`, `colsample_bytree`

- **LightGBM**: `num_leaves`, `learning_rate`, `n_estimators`, `feature_fraction`, `bagging_fraction`
- **LSTM**: number of layers, hidden units, dropout rate, learning rate, batch size, window size
- **RNN**: number of layers, hidden units, activation function, dropout, learning rate, batch size
- **N-BEATS**: stacks, blocks, hidden size, learning rate, batch size
- **N-HITS**: stacks, blocks, hidden size, pooling sizes, learning rate

All neural models are trained with Adam optimizer, early stopping with patience of 10 epochs, and learning rate reduction on plateau. Training uses a batch size of 32 and maximum of 100 epochs.

The experimental framework ensures reproducible results through fixed random seeds and comprehensive logging of all hyperparameters and results.

5 Experiments and Results

This section presents the experimental setup and comprehensive evaluation of sales forecasting models. We assess the performance of multiple approaches including traditional statistical methods, machine learning models, and deep learning architectures.

5.1 Experimental Setup

The dataset was divided into training, validation, and testing sets using a 70-20-10 split ratio to ensure robust evaluation. All models were implemented using Python with standard libraries including scikit-learn, statsmodels, and PyTorch. The experiments were conducted on a system with NVIDIA GPU acceleration to handle computationally intensive deep learning models.

5.2 Evaluation Metrics

We employed three standard metrics to evaluate forecasting performance:

- **MAE (Mean Absolute Error)**: $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- **MAPE (Mean Absolute Percentage Error)**: $\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$
- **RMSE (Root Mean Square Error)**: $\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

5.3 Results and Analysis

Tree-based Models LightGBM demonstrated strong performance with stable predictions across different time periods. As shown in Figure 3, the model effectively captured the overall trend while maintaining reasonable error margins. The gradient boosting mechanism allowed it to handle complex non-linear patterns in the retail sales data.

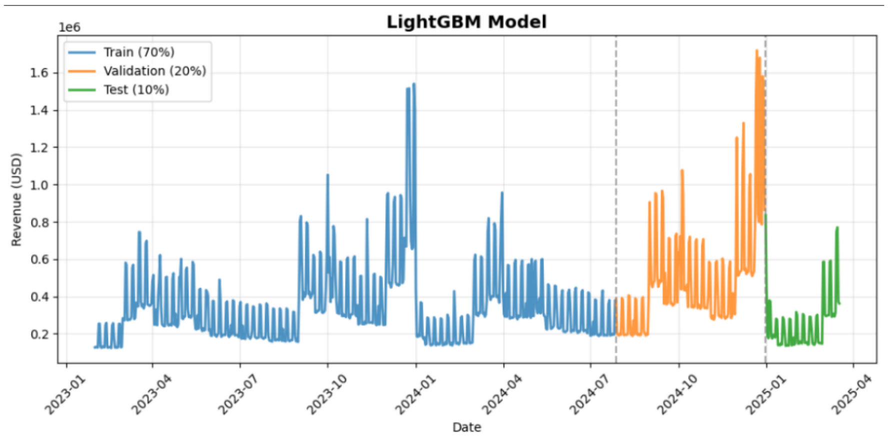

Fig. 3: LightGBM model architecture and feature importance

XGBoost exhibited comparable performance to LightGBM, with the ensemble approach providing robust predictions. Figure 4 illustrates the model's ability to track price movements accurately, though with slightly higher variance during volatile periods.

Random Forest provided stable but conservative predictions, as shown in Figure 5. The bagging approach reduced overfitting but may have limited its ability to capture rapid price changes.

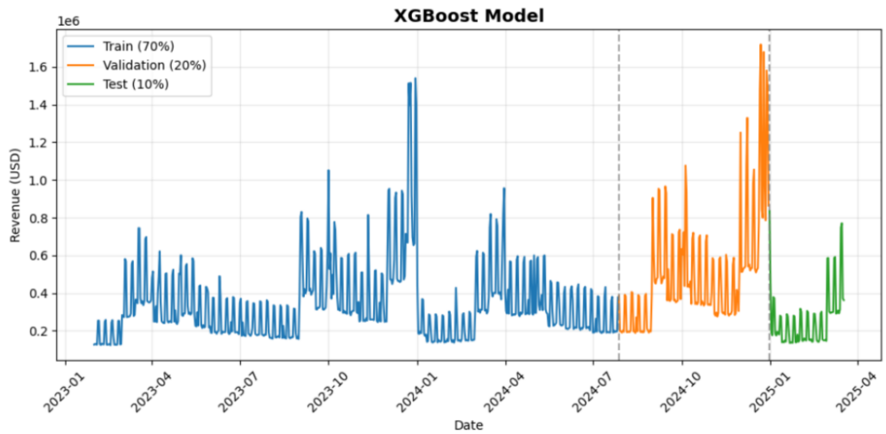

Fig. 4: XGBoost forecasting performance on test dataset

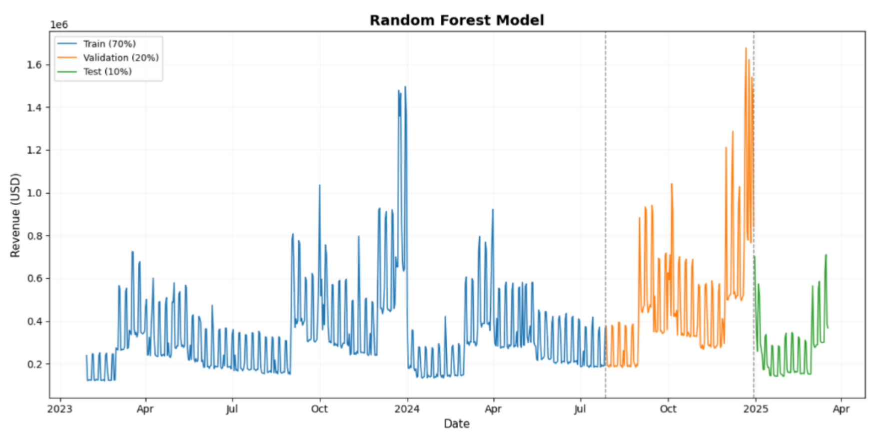

Fig. 5: Random Forest model predictions (sMAPE excluded from evaluation)

Deep Learning Models N-BEATS (Neural Basis Expansion Analysis for Time Series) showed impressive performance with its interpretable deep learning architecture. Figure 6 demonstrates the model's capacity to learn seasonal patterns and trends effectively.

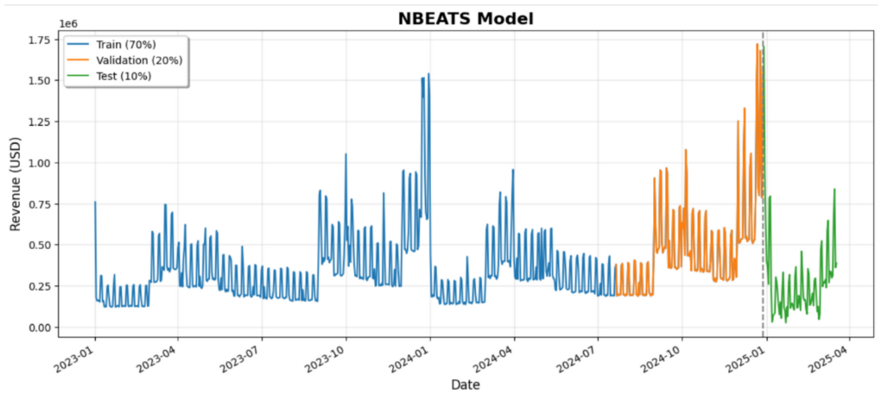

Fig. 6: N-BEATS model architecture and block design

N-HiTS (Neural Hierarchical Interpolation for Time Series Forecasting) performed comparably to N-BEATS, with its multi-rate processing enabling efficient capture of both short-term and long-term dependencies (Figure 7).

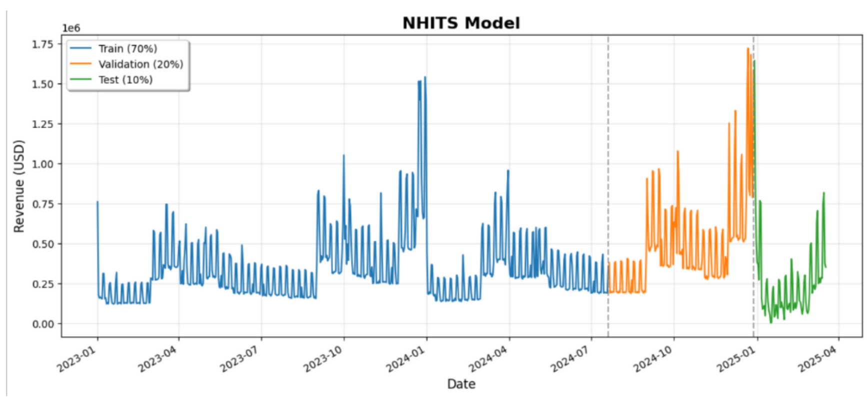

Fig. 7: N-HiTS model architecture and forecasting results

RNN and LSTM models demonstrated the capability of recurrent architectures for sequential data modeling. The LSTM (Figure 9) particularly excelled in capturing long-term dependencies, while the basic RNN (Figure 8) showed competitive performance with simpler architecture.

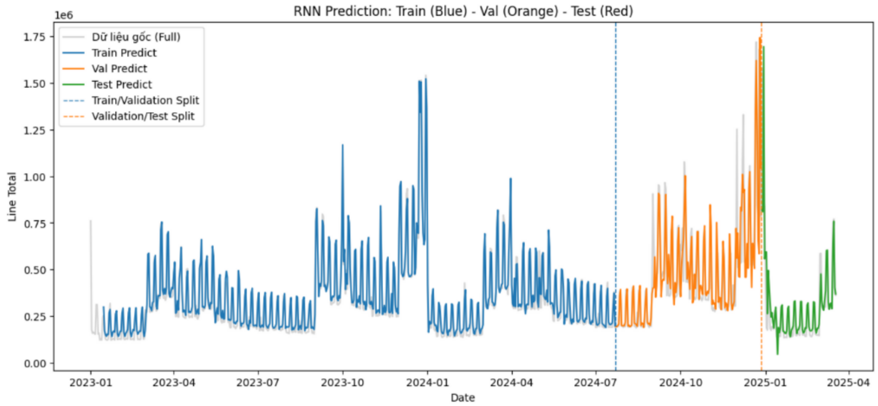

Fig. 8: RNN model forecasting results

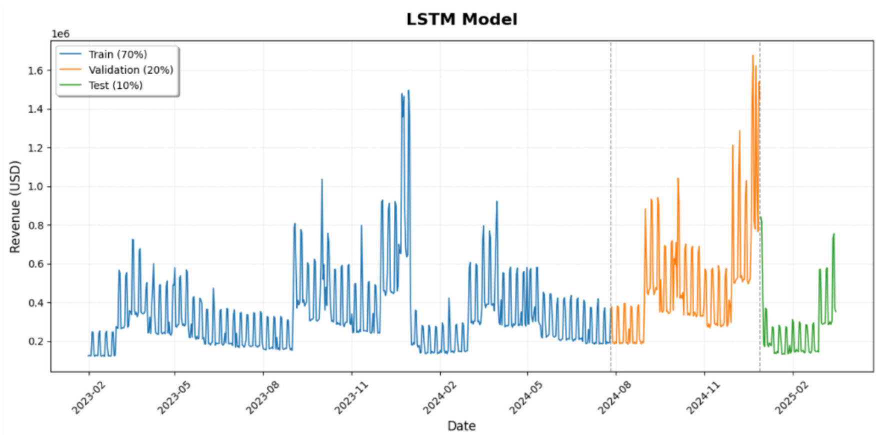

Fig. 9: LSTM model predictions showing excellent trend capture

Statistical Models SARIMA and **SARIMAX** models provided baseline performance with their well-established statistical foundations. While these models demonstrated interpretability, they generally showed higher error metrics compared to machine learning and deep learning approaches, particularly in handling non-linear patterns and abrupt market changes.

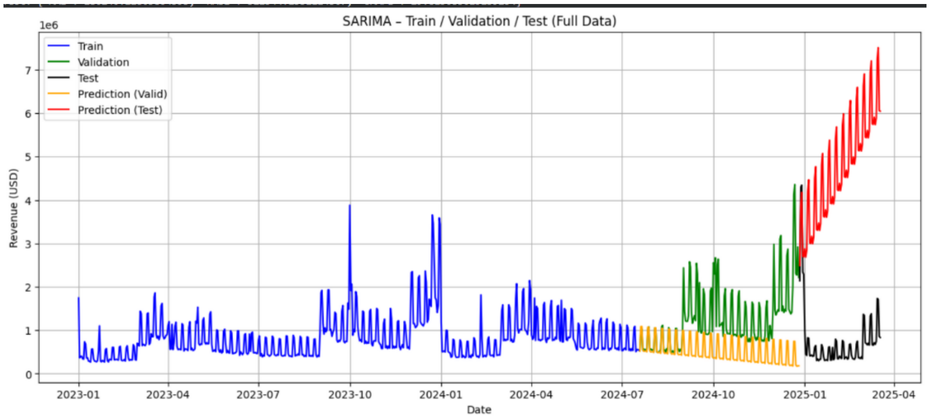

Fig. 10: SARIMA forecasting results on training, validation, and test data.

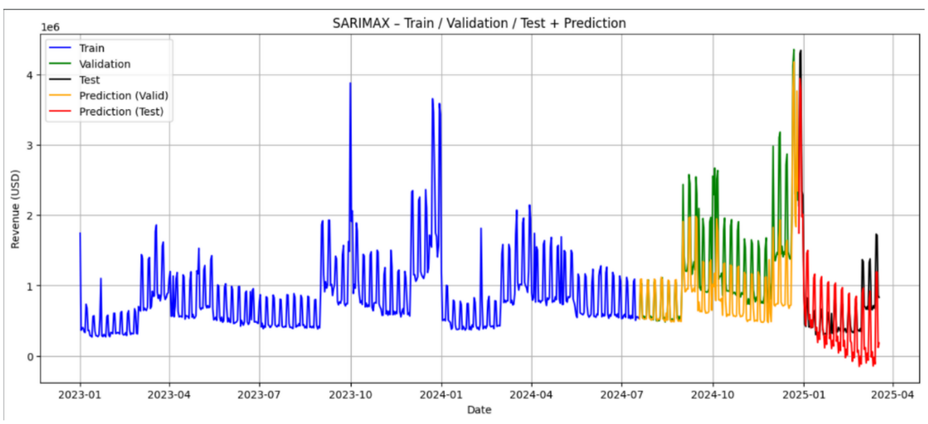

Fig. 11: SARIMAX forecasting results on training, validation, and test data.

5.4 Comparative Analysis

The experimental results reveal several key insights:

- While deep learning models are capable of capturing complex temporal dependencies, their performance in this dataset was lower than that of the tree-based models, with significantly higher MAE and RMSE values.
- Tree-based models (LightGBM, XGBoost) achieved the best overall accuracy, offering strong predictive performance, faster training times, and reasonable interpretability.

Table 2: Summary of forecasting model results across different Train–Test–Validation splits.

Model	**Train–Test–Val**	**MAE**	**MAPE**	**RMSE**
LightGBM	70–20–10	19409.89	7.56%	38627.28
XGBoost	70–20–10	23535.62	9.44%	37240.26
NBEATS	70–20–10	61474.32	25.79%	95000.39
NHITS	70–20–10	69702.95	31.15%	99925.15
SARIMA	70–20–10	3996775.81	800.44%	4209376.21
SARIMAX	70–20–10	378524.05	65.78%	459813.37
RNN	70–20–10	73675.22	22.40%	165558.37
LSTM	70–20–10	48798.98	17.04%	89224.20
Random forest	70–20–10	33458.88	14.21%	54122.80

– ETraditional statistical models (SARIMA, SARIMAX) provided interpretable results but were less effective in modeling nonlinear dynamics compared to machine learning approaches.

Overall, these findings suggest that ensemble tree-based methods are the most suitable models for this retail sales forecasting task, offering the best balance between accuracy, robustness, and computational efficiency.

6 Conclusion

This study evaluated multiple forecasting approaches—including statistical models, machine learning methods, and deep learning architectures—on a large-scale global retail sales dataset. The results consistently show that tree-based ensemble models, particularly LightGBM and XGBoost, achieved the highest predictive accuracy across MAE, RMSE, and MAPE. Their ability to model nonlinear patterns, combined with efficient training and strong generalization, makes them the most effective choice for this sales forecasting task.

In contrast, deep learning models such as N-BEATS, N-HiTS, RNN, and LSTM did not outperform the boosting-based models, producing higher error metrics despite their capacity to learn hierarchical and long-term dependencies. These results suggest that neural architectures may require additional hyperparameter tuning, more extensive historical data, or architectural refinements to achieve competitive performance in this context.

Traditional statistical methods (SARIMA, SARIMAX) delivered interpretable but less accurate forecasts, struggling to capture the nonlinear variability present in the sales time series.

Overall, the findings highlight that ensemble tree-based models offer the best balance between accuracy, robustness, and computational efficiency for retail sales forecasting. Future work may focus on hybrid approaches that combine boosting-based models with deep learning techniques, as well as the development of real-time forecasting systems to support operational decision-making and promotional planning.

References

1. Nixtla: NeuralForecast: Deep Learning for Time Series Forecasting (2024). [Online]. Available: `https://nixtlaverse.nixtla.io/neuralforecast`
2. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In: International Conference on Learning Representations (ICLR) (2020)
3. Challu, C., Olivares, K., Oreshkin, B., Ramirez, F., Mergenthaler-Canseco, M., Dubrawski, A.: NHITS: Neural hierarchical interpolation for time series forecasting. In: Proceedings of the 40th International Conference on Machine Learning (ICML), pp. 0000–0000 (2023)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
6. Biau, G., Scornet, E.: A random forest guided tour. Statistics Surveys 10, 37–100 (2016)
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer Series in Statistics, Springer (2009)
8. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
9. Ke, G., et al.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30, pp. 3146–3154 (2017)
10. Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice, 3rd edn. OTexts (2021)
11. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control, 5th edn. Wiley (2015)
12. Seabold, S., Perktold, J.: Statsmodels: Econometric and statistical modeling with Python. In: 9th Python in Science Conference (2010)
13. Smith, T.: Pmdarima: ARIMA estimators for Python (2017). [Online]. Available: `https://github.com/alkaline-ml/pmdarima`
14. Chatfield, C.: Time-Series Forecasting. CRC Press (2000)
15. Fildes, R., Ma, S., Kolassa, S.: Retail forecasting: Research and practice. International Journal of Forecasting 38(4), 1283–1318 (2022)
16. Arunraj, N.S., Ahrens, D.: A Hybrid Seasonal ARIMA and Quantile Regression Model for Daily Food Sales Forecasting. In: Proceedings of the 2015 International Conference on Operations Research and Enterprise Systems, pp. 000–000 (2015)
17. Ganguly, P., Mukherjee, I.: Enhancing retail sales forecasting with optimized machine learning models. In: 2024 4th International Conference on Sustainable Expert Systems, pp. 000–000 (2024)
18. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M5 competition: Results, findings and conclusions. International Journal of Forecasting 38(4), 1346–1364 (2022)

19. Eglite, L., Birzniece, I.: Retail sales forecasting using deep learning: Systematic literature review. CSIMQ 33, 00–00 (2022)
20. Mansur, S., Sattar, K., Hosseini, S.E., et al.: Sales forecasting for retail stores using hybrid neural networks and sales-affecting variables. PeerJ Computer Science 11, e3058 (2025)
21. Kaggle: Global Fashion Retail Stores Dataset (ricgomes) (2024). [Online]. Available: https://www.kaggle.com/datasets/ricgomes/global-fashion-retail-stores-dataset