Smart-Inventory-Demand-Forecasting using ML

Nowshin Ara Nini
Dept. of
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
nowshin.ara.nini@g.bracu.ac.bd

Nusrat Lamya Faruk
Dept. of
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
nusrat.lamya.faruk@g.bracu.ac.bd

Thamed Ahasan Rakin
Dept. of
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
thamed.ahasan.rakin@g.bracu.ac.bd

***Abstract*—Demand forecasting remained a crucial practice within retail industry. The inventory management highly depends on the accuracy level of demand forecasting to balance the costs of holding and shortage costs. Inaccurate forecasts can result in frequent stockouts, lost sales, or excessive overstock which negatively impact operational efficiency and profitability, and customer opinion. In recent years, most retail demand forecasting studies use machine learning and deep learning models such as Random Forest, XGBoost, LightGBM, LSTM, and Transformer-based models like Temporal Fusion Transformer to predict future demand. Although these methods improve prediction accuracy, some of them rely on static or aggregated data, ignoring temporal dependencies. Moreover, many of them are complex and do not clearly connect demand forecasting with future stock planning. Hence, we need to handle retail demand forecasting smartly. This study proposed a simple machine learning–based multivariate time-series approach with proper feature engineering to predict demand using a real-world retail inventory dataset containing approximately 73k rows and 15 features. A Random Forest model is suggested to predict product demand using historical sales, pricing, inventory, seasonality, and time-based features. The model is evaluated using time-based cross-validation and outperforms a baseline forecast. Compared to a baseline demand forecast (MAE = 8.34, RMSE = 10.02), the random forest model achieves significantly lower errors, with an average cross-validation MAE of 0.58 and RMSE of 0.86, and an out-of-fold RMSE of 0.90, MAE of 0.58, and R^2 of 1.00. The predicted demand is further utilized to support inventory decisions, such as determining stock levels and reorder quantities.**

***Index Terms*—Random Forest, OOF, TimeSeriesSplit, Recommended Inventory, Reorder Quantities**

I. Introduction

Demand forecasting is a technique used to estimate future consumer demand for a product or service based on historical data, market trends, and analytical modeling [3]. Traditional forecasting approaches often presume linear relationships or rely on simplified time-series models, which struggle to capture the complex and non-linear patterns present in modern retail data. Moreover, traditional forecasting methods rely on simple statistical models. However, when it comes to complicated demand factors such as pricing dynamics, promotional campaigns, seasonal variations, weather patterns, and competitor activities, these methods fail. Recent advancements in machine learning have enabled data-driven approaches to demand forecasting that can model nonlinear relationships and interactions among multiple factors. Retail demand is influenced not only by historical sales but also by inventory levels, pricing, discounts, holidays, and temporal features such as day, week, or season. These factors interact in complex ways, making tree-based ensemble models particularly suitable for this work.

We recommend a time-aware machine learning framework for retail demand forecasting and inventory decision support. The approach initiates with exploratory data analysis to understand demand behavior and feature relationships, unveiling non-linear interactions. A Random Forest model is trained on engineered features, including lagged demand, rolling statistics, and calendar effects. To prevent temporal data leakage, model performance is evaluated using time-series cross-validation on unique dates with out-of-fold predictions, providing an unbiased estimate of generalization performance.

By employing a Random Forest Regressor within a structured machine learning pipeline (incorporating one-hot encoding for categorical variables and standard scaling for numerical data), this research aims to:

- Identify the most influential factors in product demand in a retail environment.
- Develop a predictive model that minimizes the variance between the actual units sold and the forecast demand.
- Provide actionable insights for inventory managers to reduce both stock-outs and overstocking expenses.

Beyond prediction, this work demonstrates the operational relevance of forecasting by integrating inventory control logic. The predicted demand is converted into actionable inventory metrics—such as lead-time demand, safety stock, and reorder quantities—using a rule-based policy. The results show that the Random Forest model outperforms baseline forecasts, emphasizing the value of feature engineering and time-consistent evaluation in retail demand forecasting.

II. Literature Review

In recent years, researchers have shown strong interest in applying machine learning (ML) and artificial intelligence (AI) to advance retail demand forecasting over traditional statistical methods such as ARIMA and exponential smoothing which often struggle to capture the complex and nonlinear patterns found in modern retail data. Because of this limitation, many studies now focus on deep learning, hybrid ML models, and reinforcement learning (RL) techniques to produce more accurate and flexible forecasts that can also support business decisions.

The Temporal Fusion Transformer (TFT) proposed by Lim et al. [1], designed for multi-horizon retail demand forecasting, is one of the most advanced contributions in this field. The inability of conventional models to manage nonlinear demand, external impacts, and uncertainty estimation was overcome in this study. Using historical sales data and external factors including vacations, CPI, fuel prices, and temperature, the TFT model predicts weekly retail sales. The model which used probabilistic quantile forecasting and attention mechanisms, outperformed models such as XGBoost, CNN, LSTM, and CNN–LSTM when evaluated on Walmart sales data (2010–2012) from 45 stores. The TFT maintained consistent findings throughout 5-fold cross-validation, with RMSE = \$57.9K and $R^2 = 0.9875$.

Mancuso et al. [2] offered an ensemble framework that combines Gradient Boosting and Random Forests with reconciliation constraints to make sure coherence between aggregated and disaggregated forecasts from a hierarchical predicting perspective. In order to learn hierarchical dependencies and carry out reconciliation during training, their Neural Network Disaggregation (NND) model combined CNN and MLP architectures. The model outperformed top-down and bottom-up approaches using real-world datasets, including Walmart M5 competition data, Swiss electricity demand, and Italian grocery sales, delivering up to 35% MASE reduction and 18% SMAPE improvement. The key innovation lies in the end-to-end reconciliation mechanism that eliminates post-processing.

Several studies have also compared different ML algorithms to identify the most effective techniques for retail sales forecasting. An experimental study using a Citadel POS dataset (2013–2018) containing 87,000 sales records from 32 stores was conducted by Sajawal et al. [9]. They compared Linear Regression, Random Forest, and XGBoost with ARIMA and LSTM. The lowest error (RMSE = 0.63, MAE = 0.52) was achieved by XGBoost, outperforming all others and proving the effectiveness of ensemble learning in identifying nonlinear patterns of sales. Another major problem in retail forecasting is incomplete or missing data. By using datasets from Charles Clinkard (UK) and the UK Retail Sales Index, Riachy et al. [4] identified this problem. They introduced a deep probabilistic model improved with a feature "restriction-level", which represents lockdown severity. The model illustrated the effect of better data handling on forecasting performance, by achieving 10–15% RMSE reduction compared to baseline LSTM when tested on LSTM and TCN architectures.

Researchers have also incorporated AI into inventory optimization along with forecasting accuracy. A multi-agent deep reinforcement learning (MADRL) framework—MARIOD—that combines demand forecasting and inventory control for sensor-enabled supply chains was created by Yang et al. [5]. With an 18.2% reduced forecast error, 23.5% fewer stockouts, and a 96.5% service , MARIOD outperformed TFT and H-MARL, when tested on the datasets of Dunnhumby, Favorita Grocery, and UCI Online Retail. Although being computationally efficient (12.1% faster training), MARIOD's primary lackings include limited interpretability, model complexity,and sensor data dependence.

Purnamasari et al. [10] compared various forecasting models in their SME focused study for inventory management in a leathercraft SME. They compared SARIMA, SARIMAX, LSTM, XGBoost, and Gaussian Process Regression (GPR) using three years of transactional data(2020-2022). XGBoost outperformed SARIMA (RMSE=93.13; MAPE= 85.29) with (RMSE = 55.77; MAPE = 41.18), which is the best performance. The researchers came to a conclusion that ensemble Machine Learning models perform better for short, nonlinear SME datasets, whereas traditional approaches face difficulties with data sparsity. In addition, a research by Benhamida et al. [13] emphasized on improving the Stock&Buy retail platform by addressing the issue of intermittent and cold-start demand forecasting. They developed two hybrid models using historical product-level sales data from SMEs: ClustAvg, a model based on clustering for predicting new product sales, and Comb-TSB, an improved exponential smoothing mechanism for irregular demand. These models increased mean forecast accuracy by 12–15% (RMSE = 0.043) in comparison to ARIMA.

Numerous papers have shed light on inventory management, including stock optimization, real-time decision-making, and dynamic pricing. Singhal et al. [7] gave the proposal of a Smart Retail framework joining demand prediction and price optimization. Random Forest Regressor (RFR) was used on a proprietary dataset consisting of prices, stock levels, and seasonal factors. By gaining MSE = 8.15% for demand and 1.11% for price it outperformed ARIMA and linear regression. Although the challenges with interpretability and scalability still continue, the system deduced unsold inventories by 11%. Moreover, Praveen et al. [14] emphasized on inventory control for SMEs, highlighting the issue of stock imbalance and inefficient replenishment. They developed a predictive inventory model based on XGBoost regression, which improved inventory turnover efficiency and reduced out-of-stock events by 20%. Sekhar[6] applied Gradient Boosting Regressor to improve inefficient stock optimization, accuracy, and manual inventory control, achieving R^2 of 0.87 and RSME of 1.58, though it relies on a small sample dataset and lacks time series validation.

Many research papers provide broader perspectives on retail forecasting. Haque et al. [11] showed that LightGBM performed best among Ridge, Lasso, Decision Tree, and XGBM models when evaluated on multivariate time-series models incorporating macroeconomic indicators (CPI, ICS, unemployment rate) for U.S. Walmart data. Fildes et al. [12] carried out a comprehensive review used in academia and industry across market, chain, shop, and SKU levels of retail forecasting methods. After comparing ARIMA, exponential smoothing, econometric, and ML methods, they drew the conclusion that well-tuned traditional and econometric models can obtain performance comparable to complex machine learning models. However, they addressed remaining challenges or shortcomings such as weak integration of forecasting with operational decision-making, stock-out handling, and intermittent demand. To summarize, the existing studies in retail demand forecasting focus on complicated deep learning and hybrid models, which obtain high accuracy but are computationally expensive and difficult to implement in small and medium-sized businesses. Although some of these works gave good performance on small, clean datasets, which raises concerns about their adaptation to larger or real-world datasets. In addition, many lack practical interaction with real-time inventory systems and interpretability. To address these gaps, our project proposes a simple yet effective Random Forest–based forecasting framework that predicts product demand using historical sales and inventory data. The model emphasizes accuracy, interpretability, and low computational cost, making it accessible for smaller retailers too. By integrating structured preprocessing, time-series evaluation, and direct linkage to inventory decisions, this work bridges the gap between academic forecasting research and real-world retail applications

III. Methodology

A. Dataset Description

This study employed a retail store inventory forecasting dataset provided by Chauhan. [8] It contains daily observations for multiple stores having 15 features and 73100 rows. The numerical features are inventory level, units ordered, units sold, price, discount, holiday/promotion, and competitor pricing. The categorical features are store ID, product ID, category, region, weather condition, and seasonality. The dataset covers 5 stores (S001-S005), 20 products (P0001-P0020), 5 categories (Groceries, Toys, Clothing, Electronics, Furniture), 4 regions, and daily metrics from January 2022 to January 2024.

B. Data Preprocessing

To ensure consistent preprocessing across training and evaluation, the model is implemented as a single scikit-learn Pipeline. Numeric features are imputed using the median and standardized, while categorical features are imputed using the most common category and encoded using one-hot encoding. Integrating preprocessing with modeling reduces leakage during validation and ensures reproducibility [19]. We observe from fig.2 the correlation between price and competitor price is almost proportional. Therefore, we can drop competitor prices for speeding computational trade without losing almost any data. Again, on the other hand, we could drop the demand forecast because of the correlation of the units sold. But we did not because the demand forecast is the targeted feature. We actually wrapped preprocessing inside the pipeline so that we can avoid applying differently among the folds and reduce the leakages during time-series CV.

C. Exploratory Data Analysis

We conducted Cramer's V for categorical values. Categorical association was assessed using bias-corrected Cramér's V derived from chi-square contingency tables. Cramér's V ranges from 0 to 1 . Bias correction improves stability, especially under unequal category distributions [1].

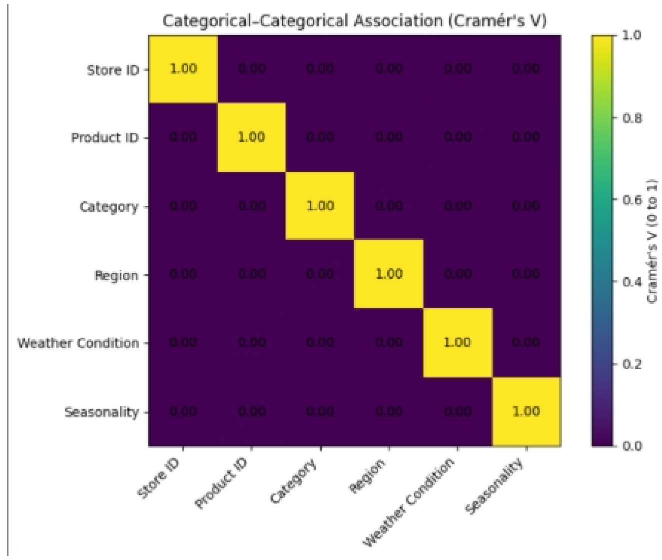

Fig. 1. Cramer's V for the categorical values.

. Now, for numerical features, we did Pearson and Kendall. Comparing both helps distinguish linear effects from monotonic but non-linear effects [16].

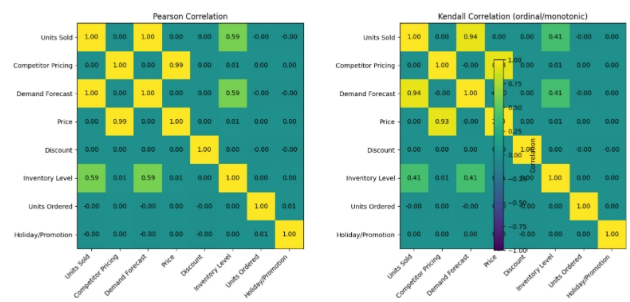

Fig. 2. Pearson is on the left and Kendall on the right .

D. Feature Engineering (time series divide)

We use Feature engineering to make the model time-aware and to capture dynamic sales. Calendar features (day of week, week of year, month, year, and weekend indicator) represent periodically. To capture demand constantly, lag features (1, 7, and 14 days) for short time and rolling statistics (7- and 30-day mean and standard deviation) for long time were computed per store-product group. The week starts on Monday. Therefore, Monday=1, Tuesday=2, and so on. Such lag/rolling constructions are commonly used in forecasting to represent autoregressive signals and evolving demand levels [18]. Inventory interaction features, including inventory gap and sell-through rate, were computed to summarize inventory behavior. For deployment-grade forecasting, any feature that depends on the same-day target should be computed using only.

E. Baseline forecast benchmark

A baseline benchmark was computed using the dataset's provided demand forecast column as the predictor and Units Sold as the actual demand. Baseline performance is reported using MAE and RMSE to provide an interpretable reference level for the ML model [21].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |actual - prediction| \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (actual - prediction)^2} \tag{2}$$

$$\text{Bias} = \frac{1}{n}\sum_{i=1}^{n} (actual - baseline) \tag{3}$$

Baseline (Demand Forecast) MAE: 8.

Baseline (Demand Forecast) RMSE: 10.01535001174343

Baseline bias (actual - forecast):-5.029850068399453

From the baseline bias, we can say that on average, the demand and sold have a difference of 5 units.

F. Random Forest Regressor (Model Description)

A Random Forest Regressor is an ML algorithm that builds multiple decision tree bootstrap trees and combines their outputs to produce a single regression prediction, which is the predicted target. Here, the collection of trees where each tree is trained through stochasticity to reduce correlation among trees and improve generalization. In regression, the final prediction is usually computed as the mean of predictions of all trees, which reduces variance and mitigates overfitting compared to a single decision tree [17]. Second, at each split within a tree, the algorithm considers only a random subset of features, which decreases the possible similarity between trees and further reduces ensemble variance [17]. Scikit-learn implementation defines Random Forest Regressor as a meta-estimator that fits decision tree regressors on various sub-samples and uses averaging to improve predictive accuracy and control over-fitting. After it is done, the result is determined by the majority or the average.

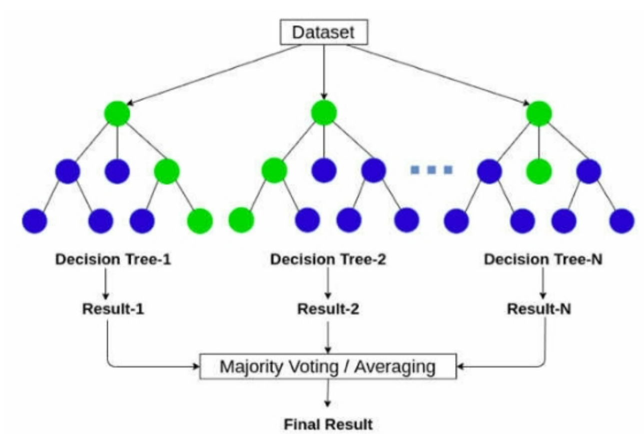

Fig. 3. Structure of Random Forest Regressor .

G. Model Implementation

The forecasting model used is a Random Forest regressor, an ensemble of decision trees trained on bootstrap samples with random feature selection. For this random selection, we can obtain different outputs, and wrong predictions can be fixed in the next tree. Random Forests reduce variance and typically perform well on tabular data with mixed feature types [17]. The model hyperparameters were set as $n_{\text{tree}} = 200$ and $max_{\text{depth}} = 12$, with parallel processing enabled.

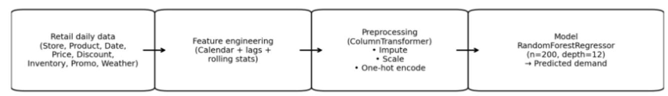

Fig. 4. Proposed machine learning pipeline for smart inventory demand forecasting.

Note: The diagram summarizes the preprocessing and Random Forest learning used in this study.

H. TimeSeriesSplit

Model evaluation uses time-awareness, where the cross-validation is checked to avoid training on future information. A 5-fold TimeSeriesSplit was applied by splitting on the unique dates so that each test fold occurs strictly after every training fold. Time series cross-validation is recommended for forecasting evaluation to avoid overly optimistic estimates [15].

I. Evaluation Metrics

Performance was evaluated using MAE, RMSE, coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE). MAE provides an average value of error, where RMSE penalizes larger errors [21].

J. Results and Discussions

Cross-validation results (5 folds) were done, and out-of-fold (OOF) predictions were aggregated to predict performance. The following fold-level results were obtained from the model run:

TABLE I
CROSS-VALIDATION PERFORMANCE OF THE FORECASTING MODEL

Fold	MAE	RMSE
Fold 1	0.89	1.32
Fold 2	0.63	0.93
Fold 3	0.50	0.75
Fold 4	0.45	0.67
Fold 5	0.43	0.63
CV Average	**0.58**	**0.86**

The OOF performance was: MAE is 0.58, RMSE is 0.90, R^2 is 1.000, and MAPE is 0.70 (Accuracy is almost 99.30%). Residual diagnostics reported a near-zero mean bias (0.01) and a residual standard deviation of 0.46 for a representative store-product series.

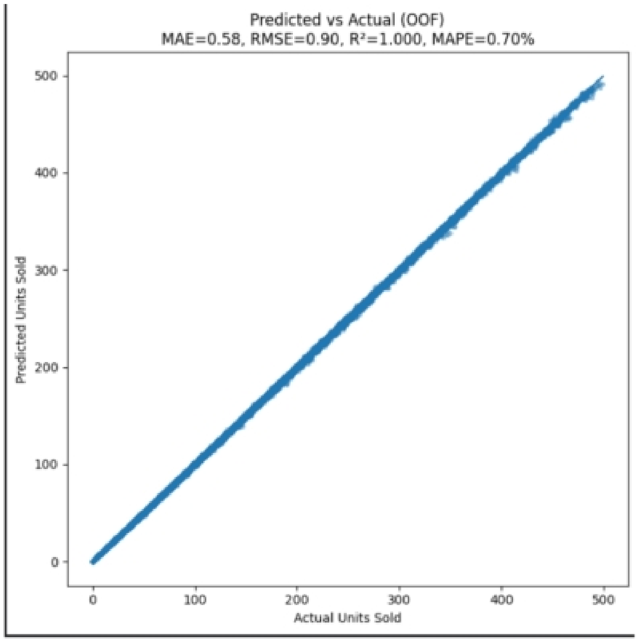

Fig. 5. Predicted vs actual demand using out-of-fold (OOF) estimates.

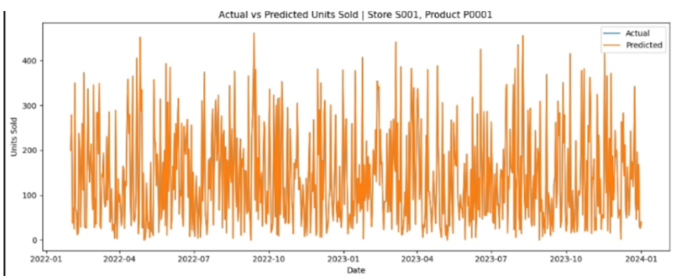

Fig. 6. Actual vs predicted Units Sold for Store S001 and Product P0001.

This figure shows a simple comparison between actual and predicted sales for one product in one store.

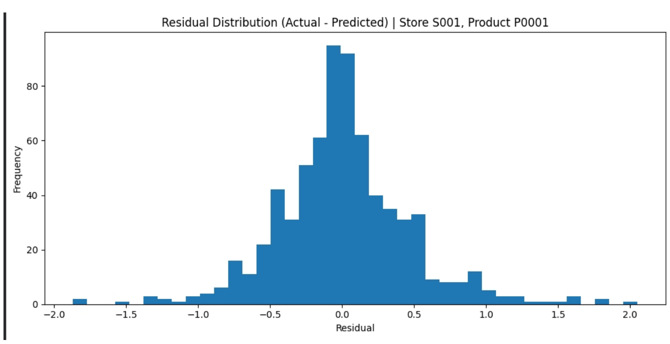

Fig. 7. Residual Distribution.

Residual analysis shows that prediction errors are centered near zero, indicating stable and reliable performance. The predicted daily demand obtained from the trained model is further utilized to make inventory replenishment decisions. The recommended inventory level is computed by multiplying the predicted demand by the supplier lead time and a service factor to incorporate safety stock.

$$I_{\text{rec}} = (\text{pred\_daily\_demand} \times \text{lead\_time\_days}) \times \text{service\_factor} \quad (4)$$

The recorder quantity is calculated as the positive difference between the current inventory level and the recommended inventory level.

$$I_{\text{reorder}} = \max(\text{recommended\_inventory} - \text{current\_inventory}, 0) \quad (5)$$

This approach ensures that replenishment is triggered only when existing stock is insufficient. The proposed logic effectively connects demand forecasting with practical inventory control. As a result, it lowers the risk of stock-outs and unnecessary overstocking.

TABLE II
SAMPLE INVENTORY RECOMMENDATION RESULTS

Date	Store ID	Product ID	Inventory Level	Predicted Demand	Recom
2024-01-01	S004	P0005	217	111.72	
2024-01-01	S004	P0006	364	194.54	
2024-01-01	S004	P0007	134	116.92	
2024-01-01	S004	P0008	390	365.35	
2024-01-01	S004	P0009	115	46.74	

IV. CONCLUSION

In short, this study implemented an end-to-end, time-aware machine learning pipeline to forecast inventory where the target (Units demand) and support smarter inventory decisions. The approach model(random forest regressor) leakage-safe preprocessing through a unified scikit-learn pipeline, and forecasting oriented feature engineering or time stamp(calendar, lag, rolling, and inventory interaction features) with baseline benchmark using the dataset's provided demand forecasting after which a Random Forest Regressor was trained and evaluated using 5-fold TimeSeriesSplit to ensure that each validation fold occurred accordingly after the training period. Model performance was assessed with MAE, RMSE, R², and MAPE, and diagnostic checks (bias and residual behavior) were used to confirm prediction stability

REFERENCES

[1] S. B. Punati, S. Kanta, U. B. Cheerala, M. G. Lanjewar, and P. Damacharla, "Temporal Fusion Transformer for Multi-Horizon Probabilistic Forecasting of Weekly Retail Sales," *arXiv preprint arXiv:2511.00552*, 2025, doi: 10.48550/arXiv.2511.00552.

[2] G. Mancuso *et al.*, "A machine-learning approach for forecasting hierarchical time series," Univ. of Rome, Italy, 2021.

[3] G. Konstantopoulos, E. P. Koumoulos, and C. A. Charitidis, "Digital innovation enabled nanomaterial manufacturing: Machine learning strategies and green perspectives," *Nanomaterials*, vol. 12, no. 15, p. 2646, 2022.

[4] J. Smith et al., "Enhancing deep learning for demand forecasting to address large data gaps," Teesside Univ., U.K., 2023.

[5] H. Al-Hassan *et al.*, "Multi-agent deep reinforcement learning for integrated demand forecasting and inventory optimization," ResearchGate, 2024.

[6] C. Veluru, "Optimizing retail inventory management with AI: A predictive approach to demand forecasting, stock optimization, and automated reordering," *Zenodo*, pp. 89–94, 2022, doi: 10.5281/zenodo.13325333.

[7] K. Singhal, V. Singh, and A. Kaul, "Smart retail: Utilizing machine learning for demand prediction, price strategy, and inventory management," in *Proc. IEEE Int. Conf. Computational Intelligence and Networks (CICN)*, 2024, pp. 485–489, doi: 10.1109/CICN63059.2024.10847534.
[8] A. S. Chauhan, "Retail Store Inventory Forecasting Dataset," Kaggle, 2025. [Online]. Available: https://www.kaggle.com/datasets/anirudhchauhan/retail-store-inventory-forecasting-dataset/data.
[9] M. Sajawal, S. Usman, H. S. Alshaikh, A. Hayat, and M. U. Ashraf, "Predictive analysis of retail sales forecasting using machine learning techniques," *J. Inf. Commun. Technol.*, vol. 14, no. 1, pp. 399–414, 2023.
[10] D. Purnamasari, V. Permadi, A. Saepudin, and R. Agusdin, "Demand forecasting for improved inventory management in small and medium-sized businesses," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 12, pp. 56–66, 2023, doi: 10.23887/janapati.v12i1.57144.
[11] M. S. Haque *et al*., "Retail demand forecasting: A comparative study for multivariate time series," *IEEE Trans.*, 2023.
[12] R. Fildes, S. Ma, and S. Kolassa, "Retail forecasting: Research and practice," *Int. J. Forecasting*, vol. 38, no. 4, pp. 1283–1318, 2022, doi: 10.1016/j.ijforecast.2019.06.004.
[13] F. Z. Benhamida *et al*., "Demand forecasting tool for inventory control: Smart systems," *J. Commun. Softw. Syst.*, vol. 17, no. 2, pp. 185–196, 2021.
[14] P. K. B *et al*., "Inventory management using machine learning," *Int. J. Eng. Res. Technol.*, vol. 9, no. 6, pp. 866–869, 2020.
[15] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics & Data Analysis*, vol. 120, pp. 70–83, 2018, doi: 10.1016/j.csda.2017.11.003.
[16] W. Bergsma, "A bias-correction for Cramér's V and Tschuprow's T," *Journal of the Korean Statistical Society*, vol. 42, no. 3, pp. 323–328, 2013.
[17] L. Breiman, "Random forests*Machine Learning*", vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[18] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., Melbourne, Australia: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/
[19] F. Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[20] E. A. Silver, D. F. Pyke, and R. Peterson, *Inventory Management and Production Planning and Scheduling*, 3rd ed., New York, NY, USA: Wiley, 1998.
[21] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, pp. 79–82, 2005.