iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.71934

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Supply Chain Accelerator for Packed Goods. A Platform for Demand Forecasting

Sai Suhas Sawant[1], Rugved Agasti[2], Nitin Sohan Jha[3]
Computer Science and Engineering D Y Patil International University, Pune, India

Abstract: With the rapid growth of online businesses, instant delivery systems, and on-demand ordering services, supply chain management has become one of the most dynamic and complex industries globally. Each day, millions of goods are transported across vast networks involving manufacturers, distributors, sup- pliers, and vendors. This intricate system, collectively known as supply chain management, is critical for ensuring timely delivery and customer satisfaction. However, despite significant techno- logical advancements, many supply chain operations still rely on outdated, heuristic-based decision-making methods. This results in persistent challenges such as overstocking, under stocking, and resource wastage, which negatively impact both efficiency and profitability .This research proposes an AI- and ML-driven web application to address these inefficiencies. By applying forecast- ing algorithms and intelligent inventory management strategies, our application aims to provide a cost-effective solution . The integration of machine learning techniques enables data-driven decision-making, resulting in better decisions by predicting the number of stocks a user must buy with a reasonable error margin.

I. INTRODUCTION

In recent years, the rapid growth of e-commerce platforms and instant delivery services has significantly transformed consumer behavior and business logistics. This change has caused significant pressure on the supply chain management industry [4], a system responsible for the coordination of goods and services. Millions of products are transported daily through intricate networks that involves suppliers, warehouses, distribution centers, and retailers. The need to deliver the right product to the right place or people at the right time has be- come extremely crucial and can give a competitive advantage to businesses [2]. Despite the important role SCM plays in the global economy, many organizations still use outdated methods for decision making, stocking, and risk management especially for small to medium business . Most solution that are out in the market are not accessible for small-scale vendors, due to various reasons like software cost, licensing, and insufficient data. We have tried to give them as much support as possible in decision making by providing a cost effective and simple yet robust application that guides them to making better decisions. Our solution leverages cutting-edge technologies and powerful ensemble machine-learning models, such as gradient boosting—to deliver accurate, scalable predictions. Fully cloud deployed, it provides secure, on-demand access to insights from anywhere in the world.

II. RELATED WORK

Artificial Intelligence (AI) is revolutionizing the sup- ply chain industry by enhancing efficiency, resilience, and decision-making across various functions. In demand forecast- ing, machine learning models analyze historical data, seasonal patterns, and external factors to predict demand accurately, reducing overstocking and stockouts [3]. AI-driven inventory management systems enable real-time stock monitoring and automated replenishment, while logistics and route optimiza- tion algorithms minimize delivery times and fuel consumption [6]. Within warehouses, AI-powered robotics and computer vision automate picking, packing, and sorting tasks, increasing speed and accuracy. Predictive maintenance solutions use sensor data and anomaly detection to prevent equipment failures, while quality control leverages deep learning for real time defect detection. AI also supports supplier risk management through multi source data analysis and natural language processing, enhancing supply chain transparency [3]
Additionally, digital twins simulate entire supply networks to test disruptions and optimize planning, and AI-powered chatbots improve customer service. However, the benefits of AI in supply chains are largely concentrated among large corporations like Walmart and Amazon, which have the fi- nancial and technological resources to implement and scale such advanced systems. Small and medium sized enterprises (SMEs) often face barriers due to high costs, lack of technical expertise, and limited access to data infrastructure [7], leading to a growing digital divide. This disparity risks marginaliz- ing smaller players and creating an uneven playing field in global supply networks. Overall, while AI holds transformative potential, broader accessibility and equitable adoption remain critical challenges for inclusive progress in the supply chain sector [2].

III. SYSTEM OVERVIEW

This web-based application presents a comprehensive full- stack architecture designed to facilitate data interaction and prediction. The frontend is developed using Next.js, a mod- ern React-based framework that enables efficient server-side rendering (SSR) and static site generation (SSG). Through Next.js, users are provided with an intuitive and interactive interface that allows them to submit data, query the system, and view prediction results in real-time. The system's frontend interacts directly with the backend through API routes, ensur- ing seamless communication and enhancing performance. On the backend, the application is powered by Spring Boot, a widely-used Java-based framework known for its scalability and robust support for creating production-grade REST APIs. The Spring Boot server is responsible for handling user input, processing the data, and managing the communication between the frontend and the machine learning model. It performs essential tasks such as data validation, preprocessing, and the orchestration of model predictions, ensuring that the system is both efficient and secure. At the heart of this application lies an AI/ML model that predicts outcomes based on user-provided data. This model, which can be trained using various machine learning algorithms, processes the data input by the user to generate predictions. The model is integrated within the Spring Boot backend, either by invoking an external service via RESTful APIs or by embedding the model directly within the backend using tools such as Python-based microservices. The backend is responsible for receiving the cleaned and processed data from the frontend, passing it to the model for prediction, and subsequently returning the results back to the frontend for display. This architecture ensures a seamless interaction between the user interface, backend services, and predictive model, making it a robust full-stack solution that empowers users to leverage AI-driven insights with ease. Through this integration of modern web technologies and machine learning, the system offers a powerful platform for data-driven decision- making.

IV. METHODOLOGY

A. Data Collecting and Processing

We began by collecting data from a variety of sources, including online repositories such as Kaggle. Additionally, we reached out to small-scale suppliers, vendors, and distributors to request access to their data. After gathering data from these diverse sources each specializing in different product cate- gories, we organized it into subcategories such as electronics, groceries, beverages, and raw materials. We further classified the data based on temporal metrics, including manufacturing date, expiry date, shelf life periods, and inventory metrics such as stock levels and shelf life. Other key metrics included lead time, supplier reliability, and warehouse location. Financial metrics such as price, transport cost, and profit margin were also recorded. The data was preprocessed through cleaning, handling of missing values, normalization of timestamps, and feature engineering. Engineered features included lead time variance, delivery accuracy, and inventory turnover rate. This structured and refined dataset served as the input for our predictive models. Notably, the data also exhibited seasonal patterns, which play a crucial role in forecasting and are discussed in detail in a later section.

The data processing workflow involved four major stages: validation and cleaning, derived variable creation, data en- richment, and quality assurance. During data validation and cleaning, we ensured the completeness of records across all fields, validated date formats and temporal consistency, con- firmed logical relationships between interdependent variables, and standardized categorical variables. In the derived variable creation phase, we calculated shelf life from manufacturing and expiry dates, determined days to expiry relative to the current date, computed stock coverage days based on cur- rent inventory and demand, and generated profit margin and revenue potential metrics. Additionally, we created transport- to-price ratios for cost analysis. The data enrichment process involved incorporating seasonal indices to account for demand fluctuations, adding historical demand data for trend analysis, linking stock levels to demand patterns, and analyzing price elasticity in relation to expiry time frames. Finally, quality assurance procedures were implemented to ensure all profit margins remained positive using cost adjustment algorithms, verify that transport costs were within reasonable bounds relative to product value, confirm that shelf life values matched industry standards for each product category, and validate that demand patterns reflected realistic seasonal variations.

B. Data insights and patterns

Data Insights and Patterns: The dataset revealed several important patterns across product categories. Category-specific shelf life patterns indicated that electronics typically have extended shelf lives ranging from 2 to 5 years, influenced more by market obsolescence than by physical degradation. Grocery items showed the highest variability, from short-lived dairy products (3–5 days) to long-lasting canned goods (1–3 years). Pharmaceuticals generally adhered to regulatory standards, with shelf lives between 1 and 3 years.

Pricing dynamics highlighted systematic reductions for products nearing expiry, particularly in the final 30% of their shelf life. These reduc- tions followed a progressive discount structure: 10% at 30% remaining shelf life, 30% at 20%, and 50% at 10%. Transport cost factors varied notably by category, ranging from 3–8% of the product price for electronics to 15–25% for refrigerated groceries. Regional differences were also observed, with trans- port costs in northern regions being approximately 10% higher than in southern regions. Seasonal demand fluctuations were prominent, with electronics demand increasing 30–60% during winter, beverages rising 40–70% in summer, and raw materials experiencing a 20–50% boost during spring and summer growing seasons. Stock management insights showed that raw materials maintained the highest stock-to-demand ratios (1.5–4× monthly demand), electronics held leaner inventories (0.5–1.5×), and pharmaceuticals maintained tightly regulated levels (0.8–2× monthly demand)

The analysis revealed various correlation patterns across product attributes, offering insights into inventory management and demand forecasting. Strong positive correlations were observed between demand and previous month demands (0.98 0.98), indicating highly consistent demand patterns, suggesting

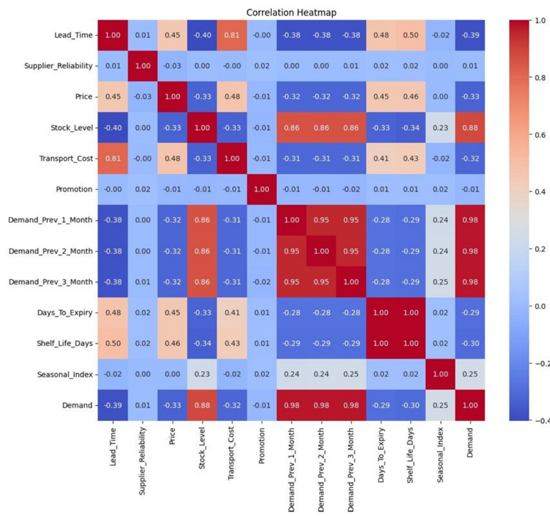


Fig. 1. Correlation Between Numerical Features in Product Demand Dataset.

strong seasonality or habitual purchasing behavior. Similarly, stock levels and demand (0.88 0.88) were closely aligned, reflecting effective inventory management practices where high-demand products are stocked in larger quantities. Lead time and transport cost (0.81 0.81) exhibited a strong positive correlation, which is logical as longer shipping times gener- ally result in higher transport expenses. A strong correlation between stock levels and previous month's demands (0.86 0.86) suggests that inventory planning relies heavily on recent sales history.

Moderate positive correlations were found between lead time and shelf life (0.48 0.50 0.480.50), as products requiring longer lead times tend to have longer shelf lives to accommodate sourcing delays. Additionally, higher-priced items exhibited moderate correlations with both lead time and transport cost (0.45 0.48 0.450.48), indicating that premium products are often more difficult to source and involve higher logistics costs.

Moderate negative correlations included lead time and de- mand (0.39 0.39), implying that products with longer lead times tend to have lower demand, possibly due to more efficient sourcing of frequently purchased items. Days to expiry and previous month demands (0.28 0.28 to 0.29 0.29) also showed a negative correlation, suggesting that products nearing expiration may see increased demand due to discounting or promotions.

Finally, no significant correlation was found with supplier reliability or promotions, as both variables showed near-zero correlation with other factors. This suggests that supplier reli- ability remains consistent across product types, while promo- tions seem to be applied uniformly across different categories without influencing demand or pricing patterns.

C. Model Training

In our research, we analyzed multiple machine learning models to identify the most accurate prediction technique for our application. The models considered for this study included Long Short-Term Memory (LSTM), Gradient Boosting, Ad- aBoost, SARIMA, ARIMA, and Random Forest.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Each of these models was evaluated based on its performance in predicting the target variable from the provided dataset. To ensure a fair comparison, we split the dataset into training and test subsets using a 70-30 split, where 70% of the data was used for training the models, and the remaining 30% was reserved for testing and validation. This division allowed us to train the models on a substantial portion of the data while retaining an unseen dataset for evaluation. The training data was used to tune the hyperparameters of each model, while the test data provided an unbiased evaluation of the model's predictive performance.

For each model, we performed hyperparameter tuning to optimize the model's performance. This process involved identifying the best combination of parameters, such as the learning rate, number of trees, and tree depth for models like Random Forest, Gradient Boosting, and AdaBoost. For time- series models like SARIMA and ARIMA, we tuned parameters such as the autoregressive order (p), differencing order (d), and moving average order (q) to find the best configuration for modeling the time-series data. The LSTM model, which requires careful tuning of layers, neurons, and learning rates, was also optimized using grid search and cross-validation techniques to find the configuration that yielded the best performance.

D. Model Formulations

Random Forest (Regression):

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$$

where T is the number of trees and $h_t(x)$ is the prediction from the t^{th} tree.

Gradient Boosting:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

where $h_m(x)$ is the weak learner, and γ_m is the learning rate.

AdaBoost (Regression):

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$$

where α_m is the weight assigned to the m^{th} learner.

ARIMA(p, d, q):

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

SARIMA(p, d, q)(P, D, Q)$_s$:

$$\Phi_P(B^s)\phi_p(B)(1-B)^d(1-B^s)^D y_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t$$

where B is the backshift operator and s is the seasonal period.

LSTM (Long Short-Term Memory):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad h_t = o_t * \tanh(C_t)$$

where σ is the sigmoid activation, C_t is the cell state, and h_t is the hidden state.

E. Model Evaluation and Performance Analysis

TABLE I
DEMAND PREDICTION MODEL PERFORMANCE METRICS

Model	MSE	RMSE	MAE	R^2
best general	220.544	14.851	9.069	0.996
general rf	220.544	14.851	9.069	0.996
general gb	407.844	20.195	14.197	0.993
general ada	1092.998	33.061	25.135	0.980
Grocery model	1999.486	44.716	27.425	0.963
Beverages model	3037.776	55.116	39.465	0.944
Raw Materials model	10497.938	102.459	74.181	0.808
Pharmaceuticals model	27679.752	166.372	88.142	0.493
Electronics model	35571.085	188.603	103.351	0.349

To evaluate the predictive performance of various regres- sion models, a comparative analysis was conducted using four key evaluation metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Er- ror (MAE), and the coefficient of determination (R^2). Among all models tested, the general-purpose Random For- est model (general_rf) demonstrated the highest overall performance, with an R^2 value of 0.99596, MSE of 220.54, RMSE of 14.85, and MAE of 9.07. This model was identified as the best general model, effectively capturing over 99% of the variance in the target variable.

Other general models, such as Gradient Boosting (general_gb) and AdaBoost (general_ada), also showed strong performance, achieving R^2 scores of 0.9925 and 0.9799, respectively, though with slightly higher error metrics.

Domain-specific models yielded mixed results. The Grocery_model and Beverages_model performed well, with R^2 values of 0.9633 and 0.9444, respectively, indi- cating strong predictive power in those categories. In contrast, models for Raw Materials, Pharmaceuticals, and Electronics demonstrated significantly lower performance, with R^2 values of 0.8078, 0.4932, and 0.3487, respectively, suggesting limited predictive capability.

These findings indicate that while generalized models like Random Forest are highly effective across datasets, further optimization is required for certain domain-specific models, particularly in the electronics and pharmaceutical sectors.

A. Model Packaging and Deployment via REST API

To enable scalable and modular access to the trained machine learning models, we packaged the best-performing models into standalone services and exposed them through RESTful APIs. These services were built using Python with the FastAPI framework due to its high performance and ease of integration with data science workflows. Each model was serialized using joblib or pickle, and loaded into memory on API startup for fast inference.

The REST API exposes endpoints such as /predict, which accept structured input data in JSON format and return predictions in real time. This approach decouples model logic from the main application backend, allowing for independent scaling and updating of models without disrupting other sys- tem components.

B. Spring Boot Backend Integration

The backend system was developed using the Spring Boot framework, which serves as a middleware layer between the frontend and the model-serving APIs. This layer is responsible for data validation, business logic execution, security handling, and interaction with the model APIs. Upon receiving requests from the frontend, the Spring Boot application processes user inputs, formats the data appropriately, and forwards it to the REST API endpoints hosting the machine learning models. Once the prediction is received, the backend handles response formatting, error handling, and persistence (if needed) before passing the result back to the user interface. In addition, the backend exposes its own RESTful end- points for frontend interaction, user session management, and database operations. This design ensures a clear separation of concerns and supports a robust, maintainable architecture.

C. Frontend Interface and User Interaction

The frontend of the application was built using modern JavaScript frameworks such as React.js and Tailwind CSS, providing a responsive and interactive interface for users. The UI allows users to input data, request predictions, view historical records, and visualize key insights through charts and tables.

Key frontend features include:

- Input forms for user-driven data submission.
- Data visualization components (e.g., bar charts, line graphs) to display model outputs and trends.
- Real-time feedback for predictions using asynchronous HTTP requests to the backend.
- Responsive design for optimal viewing across desktop and mobile platforms. The frontend communicates with the Spring Boot backend using RESTful API calls, enabling seamless interaction be- tween users and the predictive models. This layered architec- ture promotes scalability, modularity, and ease of maintenance across the entire application stack.
- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary

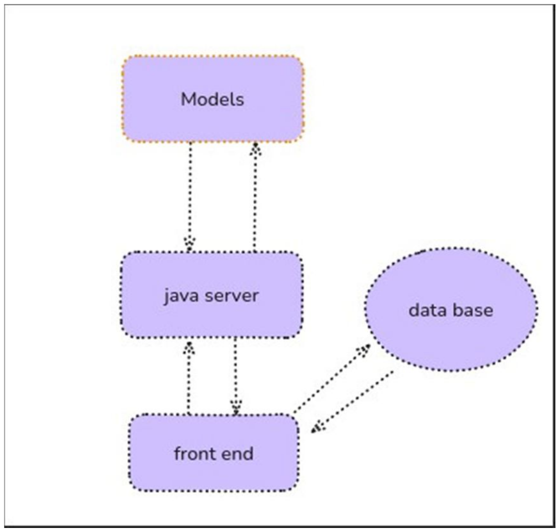


Fig. 2. Basic Overview of the system.

units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive.

- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m^2” or “webers per square meter”, not “$webers/m^2$”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “cm^3”, not “cc”.)

D. Data Collection

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

V. MODEL PERFORMANCE COMPARISON AND EVALUATION

To evaluate the effectiveness of our proposed demand fore- casting models, we compared their performance against gen- eral standards commonly observed in academic and industrial settings.

The evaluation used four key metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Ab- solute Error (MAE), and the coefficient of determination (R^2). In general practice, demand forecasting models—ranging from statistical approaches (e.g., ARIMA, exponential smooth- ing) to machine learning and deep learning techniques—are often trained on larger, more diverse datasets and benefit from

TABLE II
PERFORMANCE OF PROPOSED FORECASTING MODELS

Model	MSE	RMSE	MAE	R^2
best general (Random Forest)	220.54	14.85	9.07	0.996
general gb (Gradient Boosting)	407.84	20.20	14.20	0.993
general ada (AdaBoost)	1092.99	33.06	25.14	0.980
Grocery model	1999.49	44.72	27.42	0.963
Beverages model	3037.78	55.12	39.47	0.944
Raw Materials model	10497.94	102.46	74.18	0.808
Pharmaceuticals model	27679.75	166.37	88.14	0.493
Electronics model	35571.08	188.60	103.35	0.349

richer sets of engineered features, such as detailed seasonal trends, external market indicators, and consumer behavior data. Consequently, their performance metrics are not directly comparable to those derived from our dataset and feature set. However, they typically aim for low RMSE and high R^2 values under their respective conditions.

Despite these differences, our general-purpose Random Forest model demonstrates excellent predictive capability, achieving an R^2 of 0.996 and an RMSE of 14.85. This suggests that even with a modest feature set, the ensemble-based approach performs competitively. Gradient Boosting and AdaBoost models also yield strong results, confirming the robustness of ensemble methods in general-purpose demand forecasting. In contrast, several domain-specific models—particularly for pharmaceuticals and electronics—exhibited lower performance, with R^2 values below 0.50, suggesting that further refinements are necessary for those categories.

VI. CONCLUSION

This study set out to develop and evaluate machine learning- based demand forecasting models across various product do- mains using historical data and engineered features. Through comprehensive experimentation, we implemented and com- pared multiple ensemble models—including Random Forest, Gradient Boosting, and AdaBoost—alongside domain-specific forecasting models tailored for categories such as groceries, beverages, raw materials, pharmaceuticals, and electronics.

Our findings reveal that the general-purpose Random Forest model significantly outperforms others, achieving an R^2 value of 0.996 and an RMSE of 14.85, indicating exceptional predictive power and minimal error. Gradient Boosting and Ad- aBoost also demonstrated strong performances, with R^2 values exceeding 0.98, underscoring the effectiveness of ensemble learning in capturing complex demand patterns even without domain-specific tuning. Among the domain-specific models, those built for grocery and beverage categories performed well, with R^2 values above 0.94, suggesting that the historical features used were well-aligned with demand behavior in these segments. However, a significant disparity in performance was ob- served for models related to pharmaceuticals and electronics, where R^2 values dropped below 0.50. These results highlight potential deficiencies in feature representation, such as missing temporal trends, promotional activity, or external macroe- conomic factors, which may be more influential in these sectors. The relatively high RMSE and MAE values in these categories further point to the need for enhanced data quality and additional contextual variables. While the study demonstrates promising outcomes, several limitations must be acknowledged. Firstly, the dataset used for modeling was relatively constrained in both scope and feature richness compared to datasets used in industry-standard bench- marks. Benchmark models, often trained on larger, more di- verse datasets with access to richer features—such as customer behavior, economic indicators, and product metadata—can leverage deeper insights that were unavailable in our study. Therefore, while our results show competitive performance, particularly for general models, direct comparisons with ex- isting solutions must be interpreted with caution. Secondly, the current study relied primarily on classical machine learning approaches. Although ensemble methods have shown strong results, the exclusion of deep learning tech- niques, such as Long Short-Term Memory (LSTM) networks or Transformer-based models, may have limited our ability to capture intricate temporal dependencies, especially in volatile or highly seasonal categories. Lastly, our models assumed stationarity in the feature space and did not dynamically adapt to changes over time or exter- nal shocks (e.g., sudden market disruptions, policy changes), which can be critical in real-world demand forecasting sce- narios.

VII. FUTURE IMPROVEMENTS

While this study has demonstrated strong predictive perfor- mance in certain domains using ensemble machine learning models, several areas remain for enhancement to improve the model's generalization, robustness, and real-world applicabil- ity.

First, expanding the feature space is a critical direction for future improvement. The current model relies primarily on historical demand and a limited set of engineered features. Incorporating a wider range of contextual variables—such as promotional activity, competitor pricing, economic indicators (e.g., inflation rate, consumer sentiment), social media trends, and weather data—could provide the model with a richer understanding of demand dynamics. These additional features would be especially beneficial in volatile or seasonal markets where external factors heavily influence purchasing behavior. Second, the study's reliance on ensemble learning models like Random Forest and Gradient Boosting, while effective, does not account for the temporal nature of demand data. Future work should consider the integration of advanced time- series and deep learning techniques, such as Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), Temporal Convolutional Networks (TCNs), or Transformer- based architectures. These models are designed to capture long-range temporal dependencies, seasonality, and trends that tree-based models may overlook. A hybrid approach combin- ing deep learning with traditional ensemble methods may offer even greater forecasting accuracy.

Data enrichment and scaling also represent important av- enues for advancement. Increasing the volume, diversity, and granularity of data, particularly for underrepresented cate- gories such as pharmaceuticals and electronics, can signifi- cantly enhance model performance. Longitudinal datasets that span more extended timeframes or contain high-frequency updates can help models generalize across varying conditions. Data augmentation techniques, such as synthetic data gener- ation or bootstrapping, could also mitigate data scarcity in specific domains.

Model personalization and segmentation should be explored further. Tailoring models to specific product types, regions, or customer segments can enhance prediction accuracy, particu- larly in categories that showed lower performance in the cur- rent study. Hierarchical forecasting techniques, which provide predictions at multiple aggregation levels (e.g., SKU, category, store), can ensure consistency across forecasting layers and improve supply chain alignment.

In addition, the current model setup assumes a relatively static environment. To address real-world variability, future models should incorporate real-time data feeds and adaptive learning mechanisms. Online learning frameworks, reinforce- ment learning, or Bayesian updating can help models respond dynamically to unexpected events, such as demand shocks, supply disruptions, or policy changes. These systems would not only maintain accuracy over time but also enhance re- silience and responsiveness.

Another critical area is model explainability. As demand forecasting systems are deployed in operational settings, decision-makers require transparency to trust and act upon predictions. Incorporating explainable AI techniques—such as SHAP (SHapley Additive exPlanations), LIME (Local Inter- pretable Model-agnostic Explanations), or attention mecha- nisms—will help bridge the gap between data science and business application, enabling planners to understand the key drivers behind demand shifts.

Finally, future improvements should consider the full de- ployment pipeline. This includes building user-friendly in- terfaces, APIs, and dashboards that enable stakeholders to interact with forecasts in real time. Integration with enterprise resource planning (ERP), inventory management, and supply chain optimization systems would help operationalize the insights derived from forecasting models and generate tangible business value. In conclusion, while the current research presents a strong foundation in machine learning-based demand forecasting, significant potential remains to elevate the approach through enhanced data integration, advanced modeling techniques, personalized forecasting strategies, and scalable, interpretable deployment solutions.

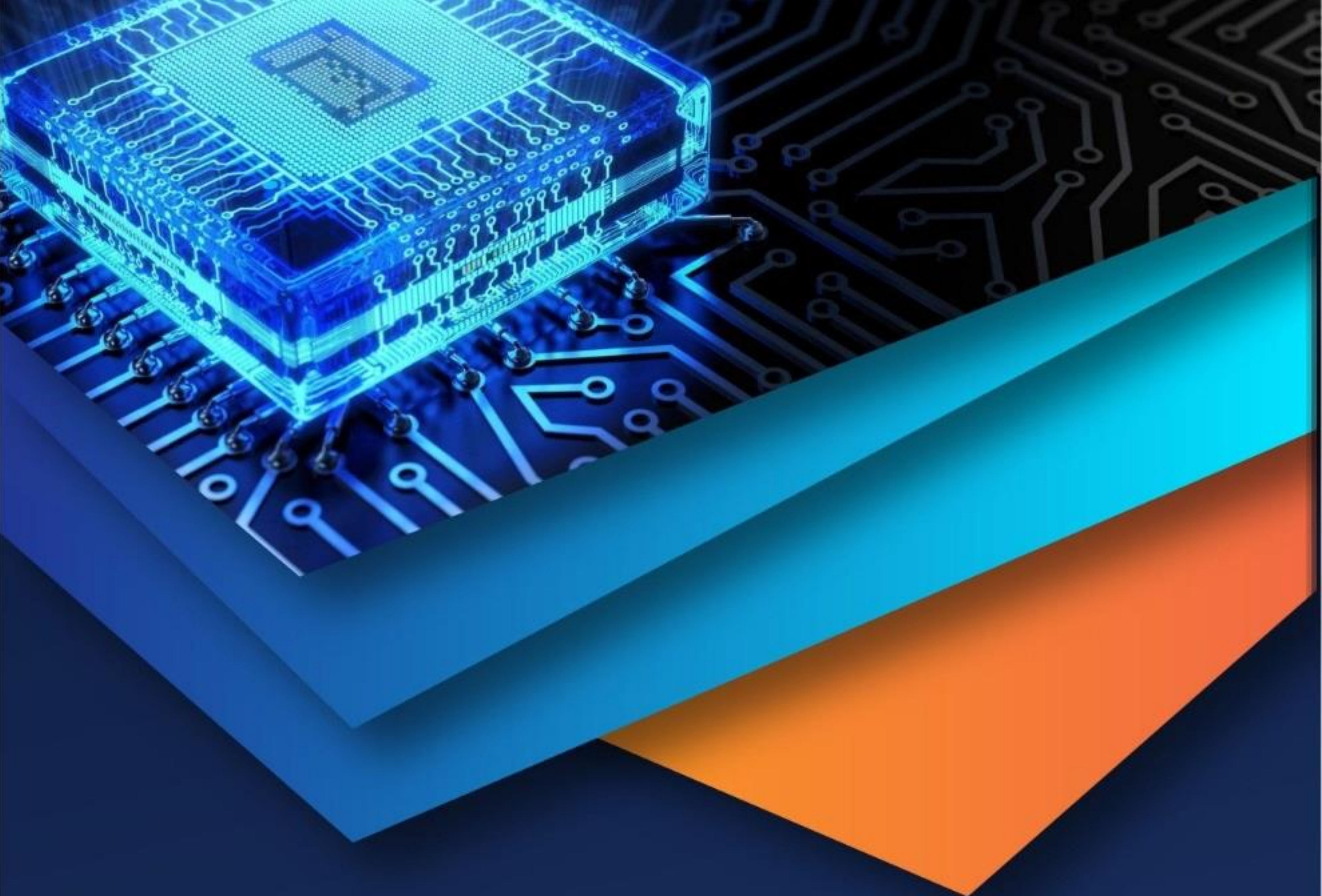
VIII. ACKNOWLEDGMENTS

We would like to express our sincere gratitude to our guide, Dr. Maheshwari Biradar, for her invaluable support, insightful guidance, and continuous encouragement throughout the course of this project. We also extend our thanks to the faculty of DY Patil International University (DYPIU) for providing the necessary resources and academic environment that enabled the successful completion of this work.

REFERENCES

[1] D. Perez, M. Gonzalez, A. Torres, and L. Ramirez, "Algorithmic approaches to inventory management optimization," *Processes*, vol. 9, no. 1, p. 102, 2021.
[2] D. Singh and A. Verma, "Inventory management in supply chain," *Materials Today*, vol. 5, no. 2, pp. 3867–3872, 2018.
[3] H. Haberleitner, H. Meyr, and A. Taudes, "Implementation of a demand planning system using advance order information," *Int. J. Prod. Econ.*, vol. 128, no. 2, pp. 518–526, 2010.
[4] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Comput. Ind. Eng.*, vol. 143, p. 106435, 2020.

[5] A.-L. Beutel and S. Minner, “Safety stock planning under causal demand forecasting,” *Int. J. Prod. Econ.*, vol. 140, no. 2, pp. 637–645, 2012.
[6] P. Sridhar, C. Vishnu, and R. Sridharan, “Simulation of inventory management systems in retail stores: A case study,” *Materials Today*, vol. 47, pp. 5130–5134, 2021.
[7] R. Granillo-Mac´ıas, “Inventory management and logistics optimization: A data mining practical approach,” *LogForum*, vol. 16, no. 4, pp. 535–547, 2020.
[8] S. Nallusamy, “Performance measurement on inventory management and logistics through various forecasting techniques,” *Int. J. Perform. Eng.*, vol. 17, no. 2, pp. 216–228, 2021.
[9] R. Carbonneau, K. Laframboise, and R. Vahidov, “Application of ma- chine learning techniques for supply chain demand forecasting,” *Eur. J. Oper. Res.*, vol. 184, no. 3, pp. 1140–1154, 2008.
[10] M. H. Naseem and J. Yang, “Role of Industry 4.0 in supply chains sustainability: A systematic literature review,” *Sustainability*, vol. 13, no. 17, p. 9544, 2021.
[11] M. Al-Okaily, H. Younis, and A. Al-Okaily, “The impact of management practices and Industry 4.0 technologies on supply chain sustainability: A systematic review,” *Heliyon*, vol. 10, no. 17, p. e36421, 2024.
[12] J. C. Quiroz-Flores, R. J. Aguado-Rodriguez, E. A. Zegarra-Aguinaga, M. F. Collao-Diaz, and A. E. Flores-Perez, “Industry 4.0, circular economy and sustainability in the food industry: A literature review,” *Int. J. Ind. Eng. Oper. Manag.*, vol. 6, no. 1, pp. 1–24, 2024.
[13] M. A. Jahin, A. Shahriar, and M. Al Amin, “MCDFN: Supply chain demand forecasting via an explainable multi-channel data fusion network model,” *arXiv preprint*, arXiv:2405.15598, 2024.
[14] S. Punia and S. Singh, “Machine learning and deep learning models for demand forecasting in supply chain management: A critical review,” *Information*, vol. 7, no. 5, p. 93, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY