IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151066

Demand & Decision Intelligence System: An AI/ML-Based Platform for Demand Forecasting, Anomaly Detection, and Inventory Optimization in Indian Retail

Prof. L.S. Bhattad[1], Yash Gawande[2], Vaishnavi Chauhan[3], Qazi Ubed Rehaman[4], Yash Zade[5], Kaushal Tayde[6], Disha Patil[7]

Professor, Department of Artificial Intelligence & Data Science, P. R. Pote Patil College of Engineering and Management, Amravati, Maharashtra, India[1]

B.Tech Student, Department of Artificial Intelligence & Data Science, P. R. Pote Patil College of Engineering and Management, Amravati, Maharashtra, India[2,3,4,5,6,7]

Abstract: Demand forecasting and inventory management help to ensure product availability and minimize inventory costs in retail or fast-moving consumer goods (FMCG). This paper introduces the Demand & Decision Intelligence System, an artificial intelligence/machine learning-based system, which combines forecasting, inventory control, anomaly detection, price elasticity, and natural language business inquiry in one web application.

The algorithm uses open-source Indian grocery data from April 1 to July 10, 2022, comprising of about 46.7 million transactions. It evaluates eight candidate algorithms, including Naive, Seasonal Naive, Moving Average, Prophet, Ridge Regression, HistGradientBoosting, Croston's Method, and Syntetos-Boylan Approximation (SBA). On the chronological holdout test set between July 1-10, 2022, the HistGradientBoosting algorithm had a weighted absolute percentage error (WAPE) of 30.08% and a mean absolute error (MAE) of 0.5481. Thus, the evaluated forecasting pipeline requires more testing to generalize the results.

Additionally, the application introduces an inventory decision support engine that automates safety stock, reorder point, and target stock calculations, a rolling Z-score and isolation forest anomaly detection module, log-log regression price elasticity model, and retrieval-augmented generation chatbot for business inquiries. The technical architecture consists of a React.js dashboard, FastAPI backend, and PostgreSQL database. Overall, the paper presents an integrated decision-support system that combines demand forecasting and inventory planning in a single platform. However, it requires additional research to evaluate the effectiveness of inventory suggestions, anomaly detection, price elasticity, and chatbot's applicability.

Keywords: Demand Forecasting, Inventory Optimization, Machine Learning, Anomaly Detection, Time-Series Analysis, Business Intelligence, RAG Chatbot, Price Elasticity, Indian Retail, FMCG.

I. INTRODUCTION

Enterprise retail operations depend heavily on demand forecasting for procurement, inventory and strategic decision-support **[1], [5].** Conventional approaches, ranging from spreadsheet heuristics and moving averages to ERP-based forecasting modules, often fail to capture nuanced demand patterns, leading to stockouts, excess inventory and increased operational costs. Stock-outs reduce sales and customer loyalty, while overstocking leads to markdowns and waste **[5].**

The Indian retail/FMCG industry has unique challenges including demand variation across multiple cities, spikes during festivals (Diwali/Holi), diverse product hierarchies (grocery, household, personal care), and multi-city logistics and supply chains. The industry needs an intelligent decision-support system that offers prescriptive analytics and addresses the unique challenges of the Indian retail ecosystem at an SKU–city level.

Recent advances in Artificial Intelligence (AI) and Machine Learning (ML) have shown improved performance over traditional approaches in forecasting and inventory management **[1], [3], [6].** However, many business intelligence platforms remain limited to reporting and lack end-to-end intelligence for inventory decision-support **[12].** Existing

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.151066

solutions also rarely combine multi-model forecasting with comparative analysis, integrated anomaly detection, price elasticity modelling and natural language querying **[9], [12].**

This paper introduces the Demand & Decision Intelligence System, an end-to-end AI/ML-based decision-support system that offers demand forecasts, inventory analytics, trend insights, anomaly detection, price–demand insights and inventory recommendations. The system is implemented as a full-stack web application using React.js, FastAPI, PostgreSQL and Python ML libraries, and was developed using about 46.7 million rows of retail transaction data.

The contributions of this paper include:

1. Multi-model demand forecasting. Development of a forecasting pipeline comparing 8 candidate statistical and machine learning models using chronological evaluation procedures.
2. Inventory decision support. Implementation of an inventory decision engine calculating safety stock, reorder points, and target stock levels using configurable service-level and lead-time parameters.
3. Demand anomaly detection. Application of Rolling Z-Score and Isolation Forest methodologies to detect outlier demand observations and assign severity categories.
4. Price-demand analysis. Implementation of a log-log regression model analyzing product-level price-demand relationships.
5. Integrated business intelligence platform. Development of React.js dashboard, FastAPI-based REST services, PostgreSQL storage, and a RAG-based chatbot providing natural-language access to business information.

The contribution of this work lies in integrating these capabilities within a unified decision-support application. The forecasting results provide an initial quantitative evaluation, while the effectiveness of the other modules requires further empirical assessment.

II. OBJECTIVES

The objective of this work is to design and implement an AI/ML-based decision support system for retail and fast-moving consumer goods (FMCG) using the provided historical data. The system will be built as a web application, utilizing several modules such as demand forecasting, inventory planning, detection of demand anomalies, analysis of price vs. demand, and business intelligence.

The work objectives include:

1. Implementing and comparing eight demand forecasting models on the given time series using chronological train/test splits.
2. Designing an inventory decision engine that will calculate safety stock level, reorder point, and target stock based on a set of customizable parameters.
3. Detecting demand anomalies using the Rolling Z-Score and Isolation Forest algorithms and implementing severity levels according to the defined rules.
4. Building a price-demand analysis module that will use a log-log regression model to calculate price elasticity of demand for products.
5. Developing an application in React.js with FastAPI backend and PostgreSQL database to visualize the results and implement a RAG chatbot.
6. Finally, demonstrating the developed system using the provided grocery database and reporting the demand forecasting module performance for the given period.

III. LITERATURE REVIEW

Demand forecasting and inventory management have been widely studied in operations research, supply chain management and, more recently, artificial intelligence and machine learning. Classical methods include linear regression, moving averages, exponential smoothing and ARIMA. These approaches are intuitive and simple to implement, but they often perform poorly when demand is nonlinear and subject to abrupt changes **[5], [6].** Recent work has applied multimodal AI to demand analysis **[2],** causal-aware transformers **[7],** large language models for intermittent demand **[10]** and time-series mixer models **[11].** Others have studied collaborative demand planning **[4],** predictive analytics for supply-chain resilience **[8]** and broader supply-chain optimization trends **[13].**

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151066

Table I: Literature Survey Summary

Sr. No.	Author & Year	Paper Title	Methodology / Technique Used	Dataset / Tools Used	Key Findings	Limitations / Research Gap
1	M. Sandra et al. (2026)	A Novel AI Analytics Driven Forecasting Paradigm for Strategic Inventory Management Through Fuzzy Multi-criteria Decision Model in Supply Chains	AI Analytics, Fuzzy MCDM (LMAW, TODIM)	Supply Chain Data	Improved inventory forecasting and decision-making using AI analytics.	Focuses on decision models rather than a practical software implementation.
2	P. Bach et al. (2026)	Adventures in Demand Analysis Using AI	Multimodal AI, Machine Learning	Text, Image and Tabular Data	AI significantly improves demand analysis and prediction accuracy.	Focuses on demand analysis; does not provide inventory management or decision support.
3	M. A. Jahin et al. (2025)	MCDFN: Supply Chain Demand Forecasting via an Explainable Multi-Channel Data Fusion Network Model	Explainable Deep Learning, Data Fusion	Multi-channel Supply Chain Dataset	Improved demand forecasting accuracy with explainable AI.	Computationally expensive and focused mainly on forecasting.
4	K. Basavaraju and O. Fatahi Valilai (2025)	Developing a Demand Planning Strategy for Joint Forecasting and Employing Analytical Tool in an Empirical Case Study	Joint Forecasting, Predictive Analytics	Industrial Case Study	Improved demand planning through collaborative forecasting.	Limited to industrial case studies; lacks intelligent recommendations.
5	A. M. Abbas et al. (2024)	Revealing the Dynamics of Demand Forecasting in Supply Chain Management	Systematic Review	Literature Survey	Discusses forecasting techniques and supply chain challenges.	Does not propose a practical AI-based system.
6	M. Muth et al. (2025)	The Application of Machine Learning for Demand Prediction Under Macroeconomic Volatility	Machine Learning, Literature Review	Research Studies	ML improves forecasting under uncertain economic conditions.	No implementation of inventory optimization or dashboards.
7	Y. Wang et al. (2025)	Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting	Transformer, Deep Learning	Text, Time-Series, Satellite Data	Achieved high forecasting accuracy using multimodal data.	Requires large datasets and high computational resources.
8	L. Bo and J. Xu (2024)	Enhancing Supply Chain Efficiency Resilience Using Predictive Analytics and Computational	Predictive Analytics, Computational Intelligence	Supply Chain Data	Improved supply chain resilience and forecasting.	Does not include conversational AI or decision intelligence.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151066

		Intelligence Techniques				
9	A. Walter et al. (2025)	Application of Artificial Intelligence in Demand Planning for Supply Chains	Systematic Literature Review	Research Articles	Highlights AI applications in demand planning and inventory management.	Primarily a review paper without implementation.
10	M. Mayr et al. (2024)	From Data to Decisions: Optimizing Supply Chain Management with Machine Learning-Infused Dashboards	Machine Learning, Business Intelligence Dashboards	Industrial Supply Chain Data	Demonstrates the value of ML dashboards for decision support.	Does not integrate forecasting, anomaly detection, and chatbot capabilities.

Research Gap Analysis: The reviewed articles address demand forecasting, inventory control, predictive analytics and supply-chain decision support from separate perspectives. Few of them combine multi-model forecast comparison, inventory planning, demand anomaly detection, price–demand analysis and natural language querying in one system. The proposed system implements these functions on a single platform with shared processing and storage. Its contribution is therefore one of implementation and integration, rather than a new forecasting algorithm.

IV. LIMITATIONS OF EXISTING SYSTEMS

Demand forecasting and inventory management encompass various activities, including forecasting future demand, detecting anomalous sales, and computing appropriate reorder quantities. If these activities are supported by different analytical or operational procedures, users may need to manually combine results. The proposed system aims to address this issue by implementing a single decision-support application.

The challenges that this work addresses are the following:

1. The analysis of time series data, reorder quantities, and demand planning may be supported by different analytical models and decision support tools, complicating the process of interpreting results and combining them into a unified action plan.
2. Depending on the characteristics of time series data, different forecasting models may yield better results; thus, the selection of an appropriate approach is critical in demand forecasting.
3. Observed sales may exhibit anomalous behavior due to various reasons, and such data must be identified and subjected to closer inspection.
4. Current descriptive analytics reports may not be sufficient in supporting the evaluation of reorder points, safety stock, and target inventory.
5. Log-transformed price and quantity can be used to estimate demand elasticity, but the observed associations may be confounded by pricing strategies, promotions and other effects.
6. Business intelligence data may be difficult to query using traditional database management systems, and natural language processing may be beneficial in this context.
7. Real-world transaction dataset, which typically originate in retail sales, must be processed, aggregated, and enriched with features to support downstream analytical tasks.

These challenges are addressed by implementing the following scope: demand forecasting, inventory decision support, anomaly detection, price-demand analysis and natural language querying. The proposed application is not designed to replace business analysts but to complement their work. This paper does not claim that commercial systems lack these capabilities; it describes an integrated open implementation of them.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151066

V. PROPOSED SYSTEM ARCHITECTURE

A. System Overview

The Demand & Decision Intelligence System's underlying architecture was built with a modular approach in mind and consists of four major blocks:

1. Data Ingestion Layer: CSV file parsing, schema validation, data cleaning, deduplication, typecasting, and ~98.86% product ID match against the product master.
2. Processing & Analytics Layer: Daily demand aggregation for product_id × city_name × date, feature engineering with DuckDB SQL, and multi-model chronological train/validation splits for forecasting.
3. Decision Intelligence Layer: Inventory optimization policy generation, anomaly detection/classification, price elasticity modeling, and recommendation of business decisions through AI.
4. Presentation Layer: React.js dashboard with Recharts data visualization, JWT-protected REST API (23 router modules), and RAG chatbot interface.

The overall end-to-end architecture that ties the blocks together is shown in Fig. 1.

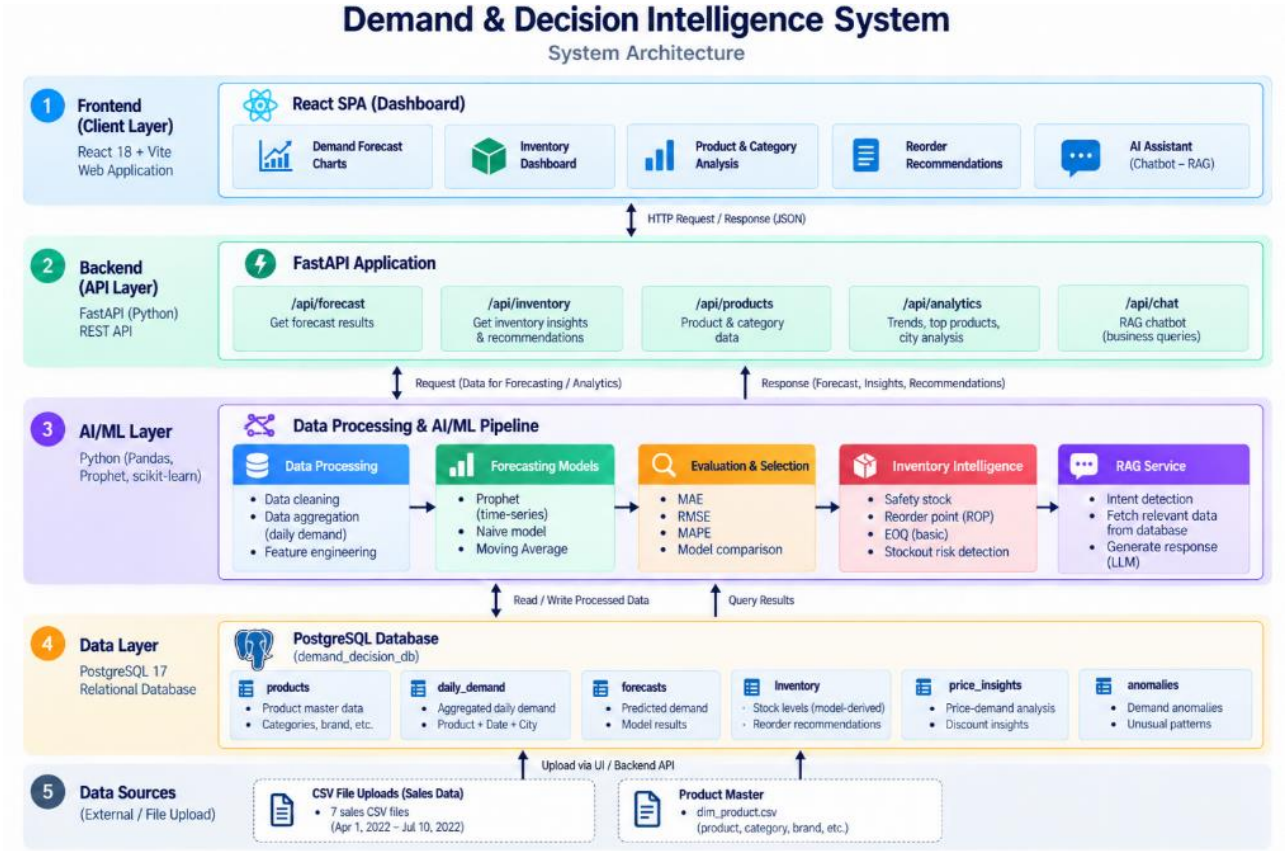


Fig. 1. Overall System Architecture of the Demand & Decision Intelligence Platform

B. Data Collection and Preprocessing

This study uses the publicly available Flipkart Grocery transaction and product details dataset, published on Kaggle by Aryan Singh **[18].** The dataset is distributed across multiple CSV files: a product master file, dim_product.csv, which provides product-level attributes, and seven sales files, about 5.9 GB in total once extracted, which contain the transaction records used for demand analysis. Together they hold roughly 46.7 million transactions from April 1 to July 10, 2022, covering about 32,226 unique product IDs in the groceries, household essentials and personal care categories. Transactions were validated and matched to the product master by product ID, with an approximate match rate of 98.86%. They were then aggregated to daily demand at the product × city × date level, with days that had no transactions recorded as zero demand. This aggregated data was used to develop and evaluate the demand forecasting and inventory decision-support pipeline.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151066

As before, state whether the 98.86% match rate is computed over unique product IDs or over transaction rows.

Table II: Dataset Summary

Parameter	Value
Total Transaction Rows	~46.7 Million
Date Range	2022-04-01 to 2022-07-10
Number of Sales Files	7 files (Extracted sales data size ~5.9 GB total)
Unique Product IDs	~32,226
Product ID Match Rate	~98.86%
Target Categories	Groceries, Household Essentials, Personal Care
Aggregation Granularity	product_id × city_name × date

The preprocessing stage involves numerous data-type validations for the required fields, including transaction date, product identifier, quantity, and selling price. Missing values and duplicates are checked before the procedure of date conversion and product-master matching. Following this step, the data points are aggregated at the level of product–city–date for demand analysis. The outliers are screened and flagged for further inspection during the exploration phase. The missing dates are matched according to the calendar, and zero demand is recorded if the transaction is absent, which implies that no sales have occurred instead of no data availability.

The product ID match rate is approximately 98.86%, according to the given implementation. The calculation basis, by unique product IDs or transaction rows, should be specified to match the response.

C. Feature Engineering

Feature engineering was performed using DuckDB SQL to enable fast analytical SQL operations. A calendar grid was created for each product-city time series to align the observations in dates. Lagged demand features, rolling features, and calendar features were engineered from the time series to showcase recent demand, demand variability, and weekly seasonality patterns in the data.

Features that predict demand included lagged demand features of size 1, 7, 14, and 28 days, rolling mean of lagged demand features of size 7, 14, and 28 days, and calendar features. The demand variability was measured using a 7-day rolling standard deviation. The calendar features included day of the week and a weekend binary indicator. Lagged and rolling features were engineered using observations that occurred before the forecast date to avoid lookahead data bias.

Then, we calculate the following features:

Table III: Engineered Features

Feature	Description	Window
lag_1, lag_7, lag_14, lag_28	Lagged demand values	1, 7, 14, 28 days
rolling_mean_7, rolling_mean_14, rolling_mean_28	Backward rolling averages	7, 14, 28 days
rolling_std_7	Backward rolling standard deviation	7 days
day_of_week	Day of the week (0=Mon to 6=Sun)	Calendar
is_weekend	Binary: 1 if Saturday/Sunday	Calendar

All rolling window calculations use strictly backward-looking windows (ROWS BETWEEN N PRECEDING AND 1 PRECEDING) to prevent lookahead data leakage.

D. Demand Forecasting Models

The forecasting pipeline evaluates eight model families (ten configurations, since the moving average is run with 7-, 14- and 28-day windows) using chronological training and validation. The first 28 days of the data are used only to build the



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.151066

lag and rolling features, so the training period spans April 29 to June 30, 2022, and the holdout period spans July 1 to July 10, 2022.

The HistGradientBoosting model was trained on a sample of 200,000 observations as specified in the implementation. The sampling procedure, random seed where applicable, and treatment of product–city time series should be documented such that the experiment can be reproduced. The evaluation should use consistent validation observations and comparable metric definitions across various models.

Walk-forward back testing is implemented in the pipeline for model selection, with training observations always preceding test observations. The final model is selected by the lowest median WAPE across completed backtest folds. The results reported in Section VI are from the single holdout window (July 1–10, 2022), not from the walk-forward folds.

The eight candidate models are:

1. Naive (Lag-1): forecast equals demand of the previous day, the simplest possible baseline
2. Seasonal Naive (Lag-7): forecast equals demand of the day-of-week of 7 days ago
3. Moving Average (7, 14, 28 days): calculate arithmetic mean over the lagging window
4. Prophet (Weekly seasonality): Facebook Prophet open source framework with weekly seasonality only enabled (yearly seasonality disabled due to insufficient historical demand for training**) [14]**
5. Ridge Regression: regularized linear regression (L2 penalty α=1.0) on the engineered feature matrix
6. HistGradientBoosting (GBT): Scikit-learn HistGradientBoostingRegressor (max_iter=60, min_samples_leaf=50) trained on a 200K sample
7. Croston's Method (Classic): intermittent demand forecasting approach which separately models demand size and time between demands (α=0.1) **[16]**
8. SBA (Syntetos-Boylan Approximation): debiased Croston variant with correction factor **[17]**

Walk-forward backtesting is employed to avoid any data leakage with train_d[-1] < test_d[0] hard constraint for chronological test folds. The champion model is selected by the lowest median WAPE across the backtest folds, and only declared if at least two models had completed the run. If the winning model failed to beat Seasonal Naive by at least 5%, Seasonal Naive is recommended with an explicit disclaimer.

E. Anomaly Detection Engine

The anomaly detection module combines a Rolling Z-Score (backward 14-day window, $|Z| > 2.5$) with an Isolation Forest **[15]** (contamination = 0.03, 100 estimators). The Isolation Forest uses three features: forecast residual, rolling volatility and sales volume. Because contamination is fixed at 0.03, the Isolation Forest flags about 3% of observations by construction.

The business logic translates to severity levels based on the value of the Rolling Z-Score, Isolation Forest's decision function, and the combination of the two (agreement). An upward direction of demand abnormality is dubbed demand spike, while the downward direction is classified as stockout-driven demand decrement. Note that a demand decrement of sales volume on its own is not enough to classify an observation as a stockout. The inventory availability information should independently confirm the suspicion of a stockout.

Anomaly detection module is designed to score observations by severity; therefore, business logic was engineered to prioritize the most critical observations. The performance of the anomaly detection module can only be evaluated on labeled events (either manually reviewed or some other form of ground truth labeling).

F. Inventory Decision Engine

The Inventory Decision Engine will estimate safety stock, reorder points, and target stock level based on the input parameters related to demand and inventory. The equations used in the current model are as follows:

- Safety Stock: $\lceil Z \times \sigma_d \times \sqrt{(L + R)} \rceil$
- ROP: $\lceil \mu_d \times L + \text{Safety Stock} \rceil$
- Target Stock Level: $\lceil \mu_d \times (L + R) + Z \times \sigma_d \times \sqrt{(L + R)} \rceil$

Here, Z is the z-score corresponding to the selected service level, σ_d is the standard deviation of daily demand, μ_d is the mean daily demand, L is the replenishment lead time in days, and R is the review period in days. The model assumes

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151066

a fixed lead time and uses the approximation σ_d × √(L + R) to estimate lead-time demand variability. The default lead time is 3 days and the review period is 7 days. The service-level z-score must be consistent with the selected service-level definition and the implementation.

It is assumed that the lead time is a fixed value, and that the estimates for demand distribution can be made with acceptable precision. The values of safety stock, reorder point, and target stock level that are produced by the equations above should be considered estimates that require adjustment for inventory position, which includes the on-hand quantity, scheduled receipts, and backorders if applicable. Additionally, the value of target stock that is calculated must be checked for optimality.

G. Price Elasticity Analysis

The price–demand module estimates item-level price elasticity by fitting a log-log regression:

$$\ln(Q) = \beta_0 + \beta_1 \times \ln(P)$$

where Q is the observed quantity, P is the observed selling price, β_0 is the intercept and β_1 is the price elasticity coefficient. Items are classified as elastic if $\beta_1 < -1$ and inelastic if $-1 \leqslant \beta_1 < 0$. Positive values of β_1 are reported separately for review, since they usually reflect promotions, availability or seasonal demand rather than a true positive price effect.

Observations with zero quantity or non-positive price are excluded because the logarithm is undefined for them. Items with too few data points or too little price variation are not estimated. Because the data are observational and prices are not set randomly, the coefficients describe association and should not be read as causal effects.

H. RAG Chatbot Architecture

The chatbot uses a four-step Retrieval-Augmented Generation process: (1) Query Understanding - understanding natural language queries and extracting intent and entities; (2) Data Retrieval - retrieving relevant forecasts, inventory records, and anomaly alerts; (3) Context Building - building structured context from retrieved data; (4) LLM Response - generating grounded natural language responses.

Table IV: Tools and Technologies

Layer	Technology	Purpose
Frontend	React 19.x + Vite	UI/Dashboard and build tooling
Frontend	Recharts	Analytics visualizations
Backend	Python 3.11+ / FastAPI	REST API and data processing
Backend	Uvicorn / Pydantic	ASGI server and validation
Database	PostgreSQL 16+	Persistent relational storage
Data/ML	Pandas / NumPy	Data manipulation and numerics
Data/ML	Scikit-learn / Prophet	ML models and forecasting
Data/ML	DuckDB	Analytical SQL processing
Auth	JWT	API authentication
DevOps	Docker / Docker Compose	Containerized deployment

VI. RESULTS AND DISCUSSION

A. Forecasting Model Evaluation Results

All of the models were evaluated on chronological validation period (2022-07-01 to 2022-07-10). Comparative results of the models are demonstrated in Table III below.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151066

Table V: Model comparison - performance metrics

Model Name	MAE	RMSE	WAPE (%)
Naive (Lag-1)	1.4522	8.5463	79.69
Seasonal Naive (Lag-7)	1.5670	8.5921	85.99
Moving Average (7 Days)	1.3211	7.3018	72.50
Moving Average (14 Days)	1.3368	7.0574	73.36
Moving Average (28 Days)	1.3757	6.9476	75.49
Prophet (Weekly Seasonality)	1.3987	9.8212	76.75
Ridge Regression	1.2041	6.3987	66.08
HistGradientBoosting (GBT)	**0.5481**	**3.0357**	**30.08**
Croston's Method (Classic)	1.4891	8.9234	81.72
SBA (Syntetos-Boylan)	1.4567	8.7891	79.93

Key Findings:

1. HistGradientBoosting had the lowest error of all candidates, with a WAPE of 30.08%, MAE of 0.5481 and RMSE of 3.0357 on the holdout period.
2. Ridge Regression (WAPE 66.08%) improved on the best simple baseline, the 7-day Moving Average (72.50%), by about 6.4 percentage points.
3. HistGradientBoosting's WAPE is less than half of Ridge Regression's. This large gap should be confirmed with a leakage check and walk-forward evaluation over multiple time windows before the result is generalised.
4. Prophet had the highest RMSE (9.8212), which suggests it handled large demand spikes poorly. Its WAPE (76.75%) was in the middle of the pack. A single Prophet configuration with weekly seasonality only may not represent its best achievable performance.
5. Croston's Method (WAPE 81.72%) and SBA (79.93%) did not beat the simple baselines on this holdout. Their results should be read together with how intermittent the evaluated series are.

These results are an initial comparison. Multiple validation windows and a documented sampling strategy are needed to confirm that the ranking holds across series and periods.

B. Anomaly Detection Results

A z-score threshold of 2.5 and an Isolation Forest contamination of 0.03 were applied, and flagged observations were ranked by severity. No labelled ground truth was available, so precision, recall and false-positive rate were not measured. These should be evaluated in future work.

However, the effectiveness analysis of the model is not provided in the case. Thus, before making any quantitative claims about the model's performance on this specific application, one would have to conduct the independent experiment, which would evaluate a model's precision, recall, false-positive rates, or other relevant metric.

C. Inventory Decision-Support Results

The Inventory Decision Engine can be used to determine safety stock, reorder points, and target stock levels for active product-city series based on the configurable inventory parameters. According to the report, the engine applies a 3-day lead time and 95% service level with a 30-day demand forecast period.

Through the provision of stock alerts and reorder recommendations, the inventory decision engine supports the decision-making process. It should be emphasized that additional analysis is required before using the engine to make decisions in the real world. This includes examining typical products and inventory results to evaluate the impact and effectiveness of the recommendations.

D. Dashboard and System Performance

The React.js dashboard contains 16 page modules for examining forecasts, inventory status, anomalies, trends, price-demand relationships, and business information. The other modules allow uploading and validating data, reviewing sales

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.151066

summaries, performing a what-if analysis, analyzing the capital allocation, suggesting purchases, processing purchase orders, planning stock and reordering, classifying products, and analyzing data quality.

The FastAPI backend consists of 23 router modules, JWT API authentication, tracing of requests via the X-Request-ID header, and Prometheus metrics. PostgreSQL serves as a warehouse for persistent relational data, while DuckDB provides analytical SQL queries and feature engineering.

These implementation details describe the system's architecture and features. They are not performance measurements. Throughput, response time and scalability have not been tested and are left for future work.

VII. CONCLUSION AND FUTURE SCOPE

This paper introduced the Demand & Decision Intelligence System, a decision-making application powered by artificial intelligence and machine learning algorithms, focused on retail and FMCG. The system was designed and developed for effective demand forecasting, inventory management, including anomaly detection, price-demand analysis, and query-based interactions through a natural language interface.

The comparison of eight candidate models in the forecasting pipeline was conducted, and in the described validation experiment on July 1-10, 2022, the HistGradientBoosting algorithm was found to have the best performance with WAPE of 30.08%, MAE of 0.5481, and RMSE of 3.0357. On this holdout window, HistGradientBoosting performed best, which suggests that gradient boosting on lag and rolling features suits this dataset. This should be tested on other periods and retail categories.

Additionally, the system includes an inventory decision engine with safety stock, reorder point, and target stock algorithms, a two-stage anomaly detection mechanism, log-log price-demand analysis model, and a RAG-based chatbot. The application's dashboard was built with React.js, while the backend API and database were deployed with FastAPI and PostgreSQL. More experiments are needed to quantify the performance of inventory management algorithms, price-demand analysis, and chatbot's response effectiveness.

The future scope of this work includes analyzing forecasting performance in different timelines and demand categories and external factors, such as festivals, weather, promotions, and prices, if the data becomes available. In addition, it might be interesting to explore the application of more advanced approaches, including real-time inventory analytics, store-to-store transfers, and model retraining pipelines. Furthermore, the RAG model's performance in terms of factual and consistent answers could be evaluated. Finally, this work demonstrated an integrated decision support system capable of automated demand forecasting, inventory management, price-demand analysis, and natural language queries. However, additional evidence is needed to evaluate its effectiveness in practical settings.

REFERENCES

[1]. M. Sandra, S. Narayanamoorthy, K. Suvitha, D. Pamucar, and D. Kang, "A Novel AI Analytics Driven Forecasting Paradigm for Strategic Inventory Management Through Fuzzy Multi-criteria Decision Model in Supply Chains," International Journal of Fuzzy Systems, vol. 28, no. 3, pp. 1094–1114, 2026.
[2]. P. Bach, V. Chernozhukov, S. Klaassen, M. Spindler, J. Teichert-Kluge, and S. Vijaykumar, "Adventures in Demand Analysis Using AI," arXiv preprint arXiv:2501.00382v3, Feb. 2026.
[3]. M. A. Jahin, A. Shahriar, and M. Al Amin, "MCDFN: Supply Chain Demand Forecasting via an Explainable Multi-Channel Data Fusion Network Model," Evolutionary Intelligence, vol. 18, Art. no. 66, 2025.
[4]. K. Basavaraju and O. Fatahi Valilai, "Developing a Demand Planning Strategy for Joint Forecasting and Employing Analytical Tool in an Empirical Case Study," Discover Applied Sciences, vol. 7, Art. no. 345, 2025.
[5]. A. M. Abbas et al., "Revealing the Dynamics of Demand Forecasting in Supply Chain Management: A Holistic Investigation," Cogent Business & Management, vol. 11, 2024.
[6]. M. Muth, M. Lingenfelder, and G. Nufer, "The Application of Machine Learning for Demand Prediction Under Macroeconomic Volatility," Management Review Quarterly, vol. 75, pp. 2759–2802, 2025.
[7]. Y. Wang, G. Ding, Z. Zeng, and S. Yang, "Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting," IEEE Access, 2025.
[8]. L. Bo and J. Xu, "Enhancing Supply Chain Efficiency Resilience Using Predictive Analytics and Computational Intelligence Techniques," IEEE Access, vol. 12, pp. 183451–183465, 2024.
[9]. A. Walter, K. Ahsan, and S. Rahman, "Application of Artificial Intelligence in Demand Planning for Supply Chains: A Systematic Literature Review," International Journal of Logistics Management, 2025.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.151066

[10]. A. Elshafie et al., "Forecasting Intermittent Demand Using Large Language Models: Evidence from a Real World Industrial Dataset," in Proc. IEEE ICTMOD, 2025.
[11]. T. N. A. Nguyen et al., "Optimizing Supply Chain Operations Using Advanced Time-Series Mixer Models for Demand Forecasting," Expert Systems with Applications, vol. 296, 2026.
[12]. M. Mayr et al., "From Data to Decisions: Optimizing Supply Chain Management with Machine Learning-Infused Dashboards," Procedia Computer Science, vol. 237, pp. 955–964, 2024.
[13]. Y. Li et al., "Frontiers and Trends of Supply Chain Optimization in the Age of Industry 4.0," Annals of Operations Research, 2024.
[14]. S. J. Taylor and B. Letham, "Forecasting at Scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.
[15]. F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in Proc. 8th IEEE International Conference on Data Mining, pp. 413–422, 2008.
[16]. J. D. Croston, "Forecasting and Stock Control for Intermittent Demands," Operational Research Quarterly, vol. 23, no. 3, pp. 289–303, 1972.
[17]. A. A. Syntetos and J. E. Boylan, "On the Bias of Intermittent Demand Estimates," International Journal of Production Economics, vol. 71, pp. 457–466, 2001.
[18]. Aryan Singh, "Flipkart Grocery transaction and product details," Kaggle. Available: https://www.kaggle.com/datasets/aryansingh95/flipkart-grocery-transaction-and-product-details. Accessed: Oct. 19, 2026.