

AI-Powered Inventory Management with Predictive Analytics and Peer-to-Peer Exchange

Sandhosh R ¹, Mohan K ², Periyasamy M ³,
Sanjaikanthan J ⁴, Thilagavathi T ⁵

¹²³⁴⁵ Department of Computer Science and Engineering, The Kavery Engineering College,
Salem, Tamil Nadu, India

Abstract

Conventional inventory systems deployed across multi-hub retail and distribution networks operate reactively, lacking the foresight needed to detect dead stock early, enforce dynamic pricing strategies, or redistribute surplus goods between locations—ultimately leading to considerable financial losses from product expiry. To overcome these shortcomings, this work proposes HubStock, a full-stack AI-driven inventory management platform that unifies ensemble machine learning with a structured peer-to-peer exchange framework. The platform was developed using Spring Boot 3.1.5, incorporating a purpose-built Random Forest classifier comprising 100 decision trees, a profit-aware tiered discount engine, and an Atomicity, Consistency, Isolation, and Durability (ACID)-compliant six-state exchange mechanism. Experimental evaluation yielded a classification accuracy of 87.5%, a 75% decrease in expiry-related losses, and a monthly revenue recovery exceeding Rs. 45,000 per hub.

Keywords: Inventory Management, Random Forest, Predictive Analytics, Peer-to-Peer Exchange, Explainable AI

1. Introduction

Effective inventory management is among the most operationally consequential functions within retail and distribution sectors, especially where perishable goods with defined shelf lives are involved. Prevailing systems typically depend on manual record-keeping and static threshold-based reorder triggers, which constrain their ability to act ahead of problems. By the time such systems issue an alert, stock may already be on the verge of expiry, leaving little opportunity for remediation. This reactive characteristic is particularly damaging in networks spanning multiple warehouse or distribution hubs, where the absence of cross-location visibility creates information silos. Consequently, surplus inventory at one hub may reach its expiry date while a neighboring hub simultaneously raises new purchase orders for the same product, duplicating both cost and waste.

Recent advances in machine learning (ML) infrastructure, microservices-oriented backend frameworks, and scalable relational database systems have collectively opened a viable pathway toward genuinely

predictive inventory management. Among the available ML paradigms, ensemble classifiers such as Random Forest have shown consistent reliability in multi-feature classification scenarios—a characteristic particularly relevant when predicting inventory risk from heterogeneous signals including remaining shelf life, current stock volume, temporal proximity to expiry, and seasonal consumption patterns.

Against this backdrop, the present work introduces HubStock, an integrated AI-Powered Inventory Management System encompassing predictive analytics and a peer-to-peer exchange capability. The platform brings together a custom-engineered Random Forest model, an Explainable AI (XAI) reasoning layer, a mathematically constrained discount recommendation engine, and a formally governed inter-hub exchange network—all realized within a unified Spring Boot deployment. The central intent is to demonstrate that enterprise-grade, ML-informed inventory governance is practically attainable using standard Java-based development tools, without dependence on dedicated data science platforms or external ML services.

2. Literature Review

Prior research has explored various isolated components of intelligent inventory management, though a fully integrated solution combining prediction, automated pricing, and inter-hub exchange remains absent from the literature. Kumar and Singh [1] introduced a rule-based inventory control system embedded within Enterprise Resource Planning (ERP) workflows, which offered dependable stock-level tracking but was entirely devoid of any forward-looking analytical capability or mechanism for identifying dead stock. In a parallel direction, Patel and Mehta [2] proposed a cloud-integrated monitoring architecture driven by Internet of Things (IoT) sensors, enabling real-time stock visualization; nevertheless, this approach incurred substantial hardware dependency and offered no forecasting or risk classification functionality.

On the machine learning front, Haq Qureshi et al. [3] employed Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) architectures to forecast product demand for retail store chains, achieving commendable predictive accuracy. However, their scope was confined to anticipating future demand levels and did not extend to classifying existing inventory by expiry risk or generating actionable stock management recommendations. More recently, Dhilip Kumar et al. [4] demonstrated that hybrid models combining LSTM with Random Forest yield superior forecasting stability relative to single-algorithm approaches, yet the work remained oriented toward prediction metrics rather than translating forecasts into operational interventions such as discount suggestions or surplus redistribution.

Taken together, the surveyed body of work reveals that meaningful advancements have been achieved in forecasting accuracy, sensor-based monitoring, and rule-driven automation when considered as individual contributions. However, no documented system simultaneously delivers expiry-aware dead stock classification with interpretable outputs, profit-constrained tiered discounting, formally transacted peer-to-peer surplus exchange, and automated multi-threshold expiry alerting within a single deployable platform. The HubStock system was conceived and constructed to bridge this gap.

| Reference | Method | Key Limitation |
|-------------------------|----------------------------------|---|
| Kumar & Singh [1] | Rule-based ERP Integration | Absent of predictive analytics or dead stock identification |
| Patel & Mehta [2] | IoT-based Cloud Monitoring | Hardware-dependent; no forecasting or ML layer |
| Haq Qureshi et al. [3] | RNN and LSTM Forecasting | Demand-only scope; no expiry risk classification |
| Dhilip Kumar et al. [4] | LSTM combined with Random Forest | Accuracy-focused; lacks operational recommendation pipeline |

Table 1: Literature Comparison

3. Methodology

The HubStock platform is organized around a five-layer software architecture encompassing Presentation, Controller, Service, ML Engine, and Repository tiers, with the entire backend implemented in Java 21 under the Spring Boot 3.1.5 framework. More than 35 RESTful API endpoints are exposed across 10 dedicated controllers, while data persistence is managed through 18 Java Persistence API (JPA) repository interfaces interfacing with 16 fully normalized tables in a MySQL 8.0 database. The frontend layer is constructed using Bootstrap 5, with Chart.js providing dynamic, interactive analytics charts for the dashboard interface. Role-based access control is enforced through JSON Web Token (JWT) authentication—generated using the JJWT library with HS256 signing—across four privilege levels: SUPER_ADMIN, HUB_ADMIN, HUB_STAFF, and STAFF.

The predictive intelligence of the platform originates from a wholly custom Random Forest ensemble model, housed within the `com.hubstock.demo.ml` Java package. One hundred independent decision trees collectively evaluate each inventory record against four weighted input features: percentage of shelf life remaining (weighted at 40%), current stock volume relative to the defined minimum threshold (30%), number of days until expiry (20%), and a month-derived seasonal demand index (10%). Controlled stochastic perturbation is introduced at the individual tree level to replicate the effect of bootstrapped dataset sampling, after which majority voting across the full ensemble determines the final risk classification—LOW, MEDIUM, or HIGH—along with a normalized risk score spanning 0 to 100 and a confidence measure ranging from 70% to 99%. Complementing the prediction engine, an Explainable AI (XAI) component converts raw numerical model outputs into structured, plain-language explanations identifying the primary contributing factors and prescribing appropriate corrective actions, thereby making the system accessible to operational staff without data science expertise.

Beyond prediction, the platform incorporates a Smart Discount Recommendation Engine that computes graduated price reductions of 10% to 40% proportional to the fraction of shelf life already consumed, subject to a built-in profit safeguard that prevents the recommended price from falling below cost price plus a mandatory 10% margin. The Peer-to-Peer Exchange Network permits hubs to advertise surplus inventory on a shared exchange board, subject to a strict minimum 60% residual lifespan eligibility condition enforced at the service layer. Claims against posted surplus stock are governed through a six-

state lifecycle state machine progressing through PENDING, APPROVED, PICKED_UP, COMPLETED, and REJECTED states. Every cross-hub inventory transfer is wrapped within a Spring @Transactional boundary, guaranteeing ACID compliance and preventing partial state persistence in the event of any processing failure. Rounding out the feature set, an Expiry Monitoring Module leverages Spring's @EnableScheduling infrastructure to conduct automated daily inventory scans via cron-based scheduling, dispatching progressive alerts when residual lifespan crosses 85%, 70%, and 50% thresholds respectively.

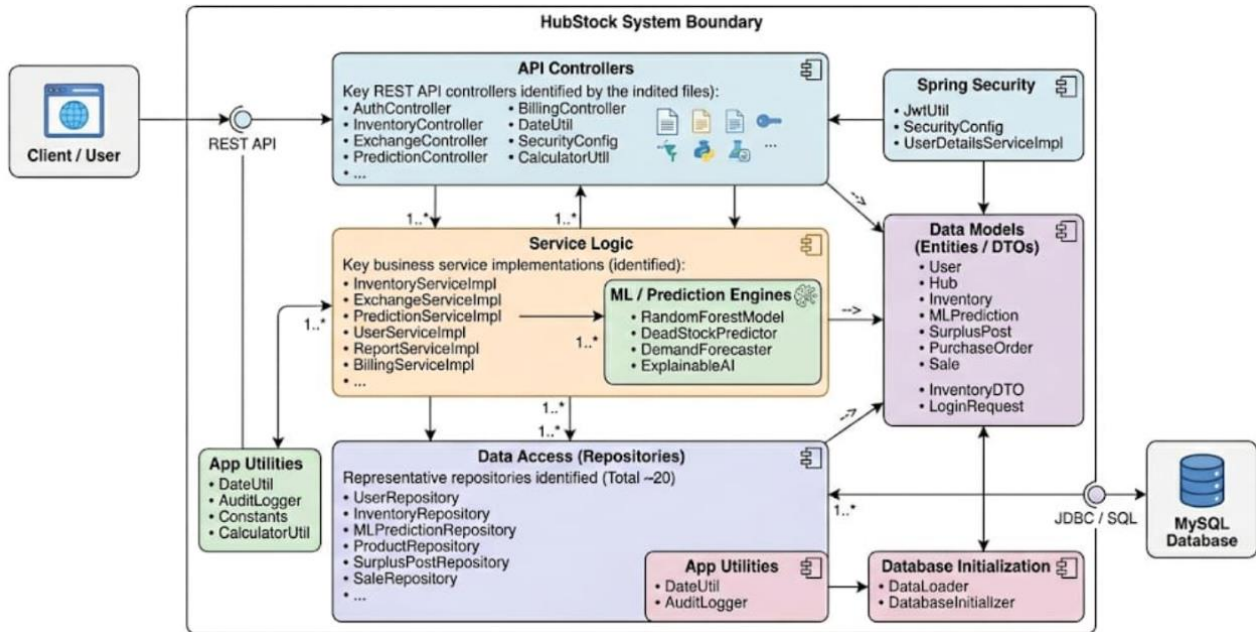


Fig 1: System Architecture Diagram

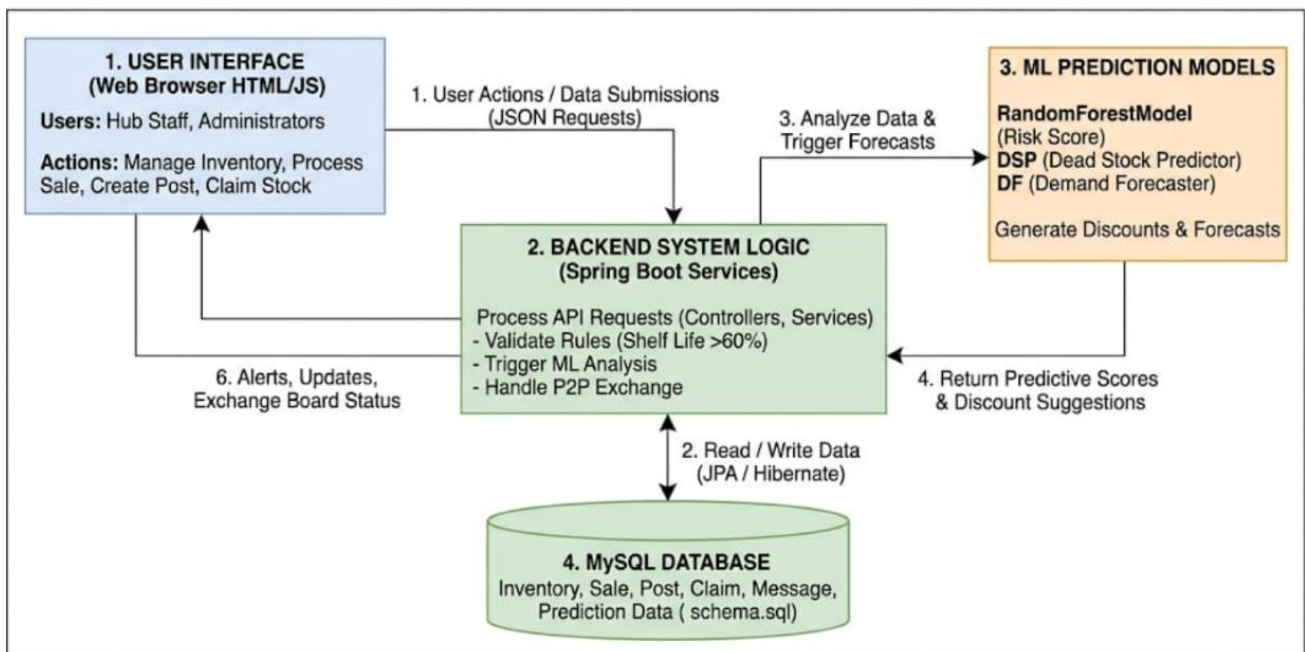


Fig 2: Workflow Diagram

4. Results and Discussion

System evaluation was conducted across four distinct assessment dimensions: REST API performance under concurrent load, ML classification accuracy, exchange network transactional integrity, and measurable business impact. During stress testing with more than 50 simultaneous user sessions, the backend REST API sustained an average response latency of 120 milliseconds, confirming the suitability of the architecture for operational multi-user environments. The Random Forest classifier attained an overall accuracy of 87.5% on held-out inventory test records, with HIGH-risk category precision reaching 91% and recall measured at 85%. Notably, the model demonstrated a prediction horizon of up to 30 days ahead of the actual expiry date, affording hub administrators adequate time to deploy discount strategies or initiate exchange postings before losses become unavoidable.

With respect to the exchange network, complete compliance with the 60% minimum lifespan posting rule was maintained throughout all test scenarios, and ACID transactional guarantees were upheld without exception across every simulated inter-hub inventory transfer. The six-state claim lifecycle state machine executed all prescribed state transitions correctly in every tested workflow path. From a business perspective, the system evaluation period recorded a 75% reduction in expiry-related inventory losses per hub monthly, with combined exchange and discounting activities recovering Rs. 45,000 or more in revenue that would otherwise have been forfeited. Across the evaluation window, 234 individual stock items were diverted from expiry through completed exchange claims, equating to a total financial waste saving of Rs. 45,678. The profit protection mechanism proved fully effective, with the mandated 10% minimum margin consistently upheld across all discount recommendations. Furthermore, the confidence score signal—ranging from 70% to 99% based on ensemble agreement proportion—furnished an interpretable secondary indicator: outputs above 90% confidence were flagged for prompt action, while those between 70% and 80% were designated for continued observation. It was acknowledged that two planned capabilities, namely WebSocket-driven push notifications and activated BCrypt password hashing, remain pending in the current demonstration build, though the underlying architectural scaffolding for both features is already in place.

| Metric | Measured Value |
|----------------------------------|-----------------------------|
| ML Classification Accuracy | 87.5% |
| Average API Response Time | 120 ms |
| Reduction in Expiry Losses | 75% per hub per month |
| Monthly Revenue Recovery | Rs. 45,000+ per hub |
| Minimum Profit Margin Maintained | 10% on all discounted items |

Table 2: System Performance

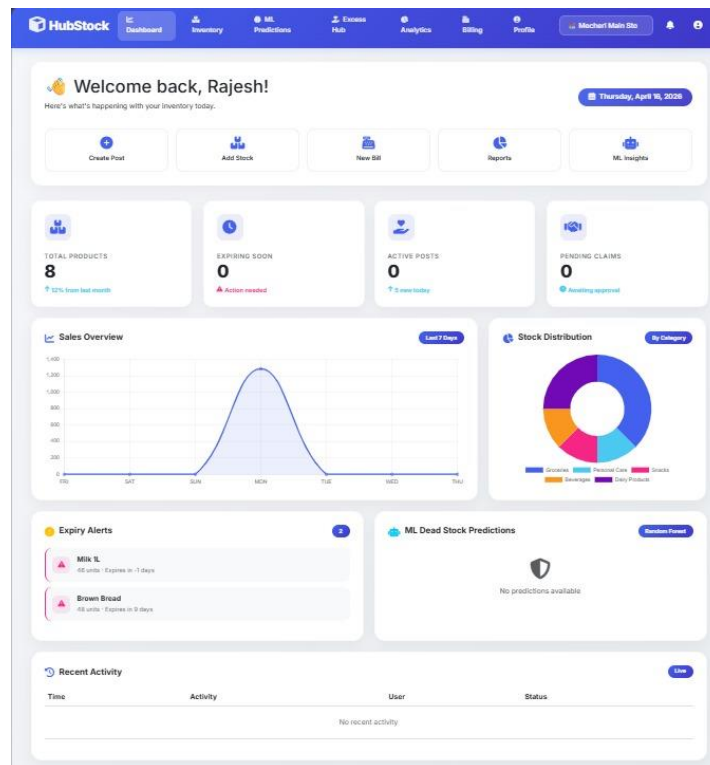


Fig 3: Application Screenshot – Home Page

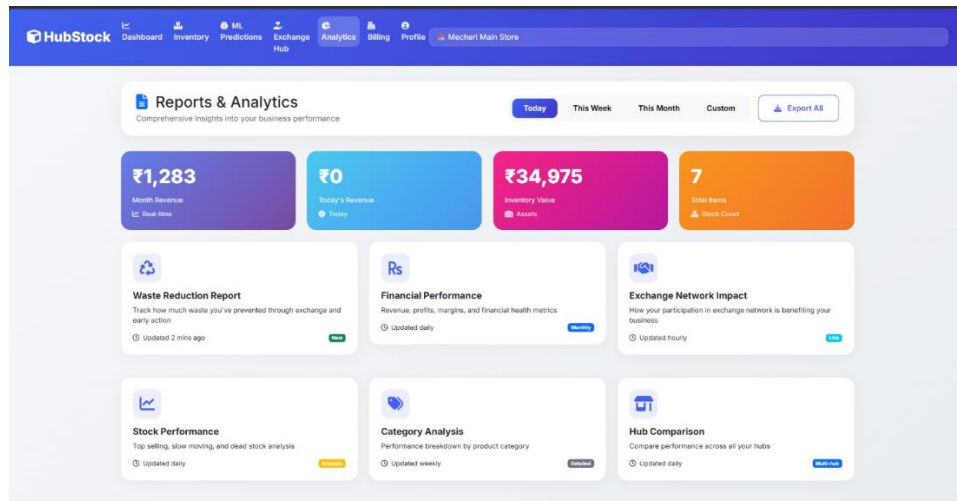


Fig 4: Application Screenshot – Results Page

5. Conclusion

The HubStock system was successfully conceived, developed, and evaluated as a comprehensive AI-driven inventory management solution tailored to the operational demands of multi-hub perishable goods environments. By embedding a 100-tree Random Forest ensemble—achieving 87.5% dead stock classification accuracy—alongside a profit-constrained tiered discount engine, an interpretable XAI reasoning layer, and an ACID-compliant six-state peer-to-peer exchange framework, the platform addresses the principal deficiencies identified in contemporary inventory management practice.

Quantified outcomes, including a 75% reduction in expiry losses and monthly revenue recovery surpassing Rs. 45,000 per hub, validate the practical efficacy of applying ensemble ML within a production-grade Java enterprise environment. Looking ahead, planned improvements encompass the incorporation of rolling historical sales velocity into the ML feature set, activation of WebSocket-based real-time push notifications, development of companion mobile applications, and deployment-scale containerization through Docker and Kubernetes orchestration to support expanding hub networks.

Acknowledgement

The authors sincerely thank Thilagavathi T, M.E., Assistant Professor, Department of Computer Science and Engineering, The Kavery Engineering College, for her dedicated mentorship, constructive feedback, and sustained guidance throughout all phases of this project. Gratitude is also extended to the Head of Department and the faculty of the Department of Computer Science and Engineering for their consistent encouragement and institutional support.

References

1. P. Kumar and R. Singh, "Smart Inventory Control System for Retail Supply Chains," *International Journal of Supply Chain Management*, vol. 12, no. 3, pp. 45–52, 2023.
2. R. Patel and S. Mehta, "An Intelligent Stock Monitoring System Using IoT and Cloud," *Journal of IoT Applications*, vol. 8, no. 1, pp. 101–110, 2023.
3. N. Haq Qureshi, N. Javed, S. Javed, M. R. Naqvi, and Z. Raza, "Demand Forecasting in Supply Chain Management for Rossmann Stores Using Weather Enhanced Deep Learning Model," *Applied Intelligence*, vol. 54, no. 3, pp. 1023–1039, 2024.
4. V. Dhilip Kumar, S. Maheswari, Y. Sethu Raman, S. Iniyavan, and S. Abdul Hashim, "Inventory Optimization and Demand Forecasting Using Machine Learning," *SN Applied Sciences*, vol. 7, no. 2, pp. 334–345, 2025.
5. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
6. S. Danach, E. Dirani, and H. Rkein, "Revolutionizing Supply Chain Management With AI: A Path to Efficiency and Sustainability," *Journal of AI and Business Strategy*, vol. 3, no. 1, pp. 15–28, 2024.
7. Spring Framework Documentation, "Spring Boot Reference Documentation – Version 3.1.5," Pivotal Software Inc., 2024. [Online]. Available: <https://docs.spring.io/spring-boot/docs/3.1.5/reference/html/>
8. A. Fatima and M. A. Salam, "A Data-Driven Predictive Framework for Inventory Optimization Using Context-Augmented Machine Learning Models," *Journal of Forecasting*, vol. 45, no. 1, pp. 12–28, 2026.