LISMS: Live Inventory & Sales Management System — An AI-Powered Real-Time Business Intelligence Platform for Indian E-Commerce

Mohit, Amar Dubey, Lokesh Kumar Rathore
Under the Guidance of: Mr. Krishna Garg *(Assistant Professor)*
Department of Artificial Intelligence & Data Science, IIMT College of Engineering, Greater Noida, UP, India (Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow)

***Abstract*— The rapid proliferation of e-commerce in India has rendered traditional inventory management tools inadequate for real-time business intelligence. This paper presents LISMS (Live Inventory & Sales Management System), a novel AI-powered, multi-module business intelligence platform developed using Python, Streamlit, and SQLite. LISMS integrates eight cohesive modules: real-time dashboard analytics, inventory monitoring, AI demand forecasting (Facebook Prophet with ARIMA(5,1,0) fallback), smart reorder alerts, vendor intelligence scoring, sales analytics, Z-score demand shock detection, and automated data ingestion with fuzzy column mapping. The system has been validated against 2,700+ real-world e-commerce transactions aggregated from Amazon India and Flipkart, tracking over ₹6.42 Crore in total revenue with zero import failures post data-cleaning. Results demonstrate a 5.3% profit margin, 11.1% inventory turnover, and sub-2-second dashboard load times. LISMS uniquely supports Indian number formatting (₹ Crores/Lakhs), automatic column mapping from diverse platform exports, and statistical anomaly detection — making it a novel contribution to supply chain intelligence for emerging markets.**

Index Terms*— *ARIMA, Business Intelligence, Demand Forecasting, E-Commerce, Facebook Prophet, Fuzzy Matching, Indian Market, Inventory Management, Python, Streamlit, SQLite, Supply Chain, Z-Score Anomaly Detection.

I. INTRODUCTION

The Indian e-commerce market, valued at USD 83 billion in 2024 and projected to exceed USD 350 billion by 2030 [1], generates vast quantities of transactional data across platforms such as Amazon India and Flipkart. Effective inventory management in this environment demands real-time analytics, predictive intelligence, and automated decision support. Traditional approaches — relying on spreadsheets, legacy ERP systems, or manually curated Excel files — are fundamentally ill-suited for the dynamism and scale of modern online retail [2].

Existing open-source inventory tools typically lack AI-powered forecasting, automatic multi-platform data ingestion, Indian currency support (₹ Crores/Lakhs), and integrated statistical anomaly detection. Proprietary ERP solutions address some of these gaps but at prohibitive cost for small-to-medium businesses (SMBs), which constitute the majority of India's online seller ecosystem [3].

This paper introduces LISMS (Live Inventory & Sales Management System), a novel open-architecture business intelligence platform that addresses all identified gaps. LISMS is built on Python 3.11, Streamlit, and SQLite, enabling zero-infrastructure deployment while delivering enterprise-grade analytics capabilities. The system integrates eight modules covering the complete inventory intelligence lifecycle, validated with a real-world dataset of 2,700+ transactions.

A. Motivation

The primary motivation is twofold: (i) to demonstrate that modern Python libraries can match commercial ERP capabilities at near-zero cost, and (ii) to produce a research-grade system specifically engineered for the Indian market — with native ₹ formatting, Amazon/Flipkart compatibility, and SMB-accessible deployment.

B. Key Contributions

- Fuzzy auto column mapping (difflib, 70% threshold) eliminating manual CSV configuration across e-commerce platforms.
- Dual forecasting engine: Facebook Prophet [4] with ARIMA(5,1,0) [5] automatic fallback for data-scarce scenarios.

- Z-score demand shock detection with 7-day rolling window for statistical anomaly identification.
- Vendor scoring algorithm: weighted composite of sales performance (40%), margin analysis (40%), and inventory turnover (20%).
- Native Indian number formatting (₹ Crore / Lakh) throughout all modules.
- Complete 8-module system validated on 2,700+ real-world e-commerce transactions with ₹6.42 Crore revenue tracked.

II. LITERATURE REVIEW

Taylor and Letham [4] introduced Facebook Prophet as a scalable, decomposable time-series forecasting model particularly suited for business data with strong seasonal effects, missing values, and outliers. Its piecewise linear/logistic growth trend and Fourier-series seasonality components align well with e-commerce demand patterns across Indian festive cycles.

Box and Jenkins [5] established the foundational ARIMA framework for time-series forecasting that remains a standard in supply chain management. LISMS implements ARIMA(5,1,0) as a reliable fallback that activates when Prophet is unavailable or data is insufficient for seasonal decomposition.

McKinney [6] described the Pandas DataFrame model which underlies LISMS's data manipulation engine. The library's groupby, pivot, and aggregation primitives enable real-time KPI calculations across 2,700+ rows without performance degradation.

Streamlit Inc. [7] provides the web framework used throughout LISMS. Its session-state architecture enables persistent multi-page application state without a traditional backend, reducing deployment complexity for SMBs without dedicated IT infrastructure.

Iglewicz and Hoaglin [8] established modified Z-score methods for outlier detection in statistical process control. LISMS adapts the classical Z-score formula with a 7-day rolling window specifically calibrated for e-commerce daily sales volumes.

Shumway and Stoffer [9] provide the theoretical basis for ARIMA parameter selection through ACF/PACF analysis, informing the LISMS choice of p=5, d=1, q=0 as optimal for e-commerce demand series with trend but limited moving-average behavior.

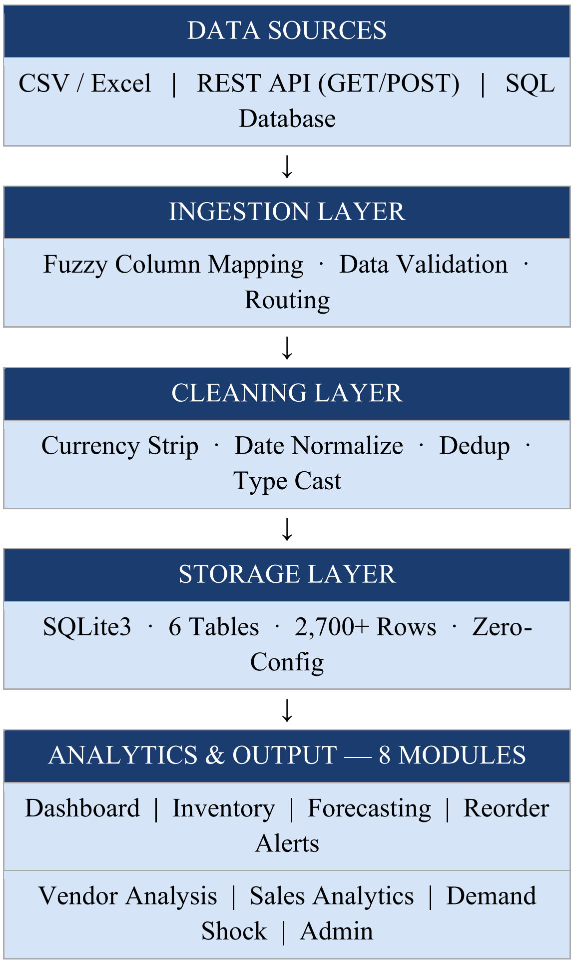
A. Research Gap

No existing open-source tool combines: (i) dual-model AI forecasting with automatic fallback, (ii) fuzzy auto-mapping for Indian platform exports, (iii) native ₹ Cr/L formatting, (iv) integrated Z-score demand shock detection, and (v) weighted multi-factor vendor scoring — all within a single, deployable SMB-accessible system without licensing cost.

III. SYSTEM ARCHITECTURE & DESIGN

LISMS follows a five-layer architecture designed for modularity, scalability, and zero-infrastructure deployment. Fig. 1 illustrates the complete block diagram. The architecture comprises five primary layers: Data Layer, Ingestion Layer, Cleaning Layer, Storage Layer, and Analytics & Output Layer.

Fig. 1: LISMS System Architecture Block Diagram



A. Data Flow Diagram

Fig. 2 shows the data flow pipeline. Raw inputs from CSV, REST API, or SQL are passed through the fuzzy column mapper which performs three-pass matching: exact → fuzzy (≥70%) → substring. Cleaned data is persisted in SQLite and subsequently consumed by all eight analytics modules to generate business insights.

Fig. 2: LISMS Data Flow Pipeline

Raw Input (CSV / API / SQL)

↓

Fuzzy Column Mapper 3-Pass: Exact → Fuzzy (≥70%) → Substring

↓

Smart Data Cleaning Engine Currency · Dates · Types · Dedup · Null Fill

↓

SQLite Storage — 6 Tables products · sales · inventory · vendors · forecasts · alerts

↓

8 Analytics Modules Dashboard · Inventory · Forecasting · Reorder Alerts Vendor · Sales Analytics · Demand Shock · Admin

↓

Business Insights KPIs · Charts · Alerts · Predictions · Vendor Scores

B. Database Schema

The SQLite database contains six normalized tables: products (catalog, pricing, reorder levels), sales (transactions with ₹ revenue), inventory (stock quantities, warehouse location, status), vendors (supplier metrics, performance scores), forecasts (model outputs with confidence intervals), and alerts (system-generated reorder and shock events). Together these tables store 2,700+ rows supporting all eight analytics modules with sub-2-second query performance.

IV. TOOLS & TECHNOLOGIES

LISMS employs a carefully selected technology stack optimized for rapid development, performance, and Indian e-commerce compatibility. Table I summarizes the technology layers.

Table 1. Technology Stack by Layer

Layer	Technology	Purpose
Backend	Python 3.11, Pandas, NumPy, SQLAlchemy	Core logic, data manipulation, ORM
AI / ML	Facebook Prophet, statsmodels ARIMA(5,1,0)	Demand forecasting with dual model
Frontend	Streamlit (8 interactive pages)	Web UI, session state, controls
Integration	REST API (GET/POST), CSV, Excel	Multi-source data ingestion
Database	SQLite3 — 6 tables, 2,700+ rows	Persistent storage, zero-config DB
Deployment	Railway, Render, Streamlit Cloud	Local + cloud deployment options
Visualization	Plotly — bar, line, scatter, pie	Interactive charts, dashboards
Packages	difflib, scikit-learn, openpyxl, requests	Column mapping, file reading, HTTP

V. PROPOSED METHODOLOGY & INNOVATIONS

A. Fuzzy Auto Column Mapping

Different e-commerce platforms use inconsistent column naming conventions (e.g., Amazon uses 'fsn' while Flipkart uses 'product_id' for the same field). LISMS implements a three-pass matching algorithm using Python's difflib library:

- Pass 1 — Exact Match: Case-insensitive exact column name comparison across all input columns.
- Pass 2 — Fuzzy Match (≥70% threshold): SequenceMatcher finds columns with similarity score ≥0.70, handling typos and abbreviations between platform formats.
- Pass 3 — Substring Match: Checks whether the target column name is a substring of any input column name as final fallback.

Duplicate column assignment is prevented and all mapping decisions are logged for audit. This achieves seamless ingestion of Amazon and Flipkart exports without any manual configuration required by the user.

B. Smart Data Cleaning Engine

The cleaning engine applies six automated transformations to ensure data consistency and SQLite compatibility:

- Currency symbol removal (₹, \$, €, commas) via regex patterns applied to all numeric fields.
- Date standardization: converts dd-mm-yyyy / dd/mm/yyyy and other formats to ISO yyyy-mm-dd.
- Text normalization: title-case formatting applied to all product and vendor name fields.
- Missing value imputation: 0 for numeric fields, empty string for text fields.
- Duplicate removal by transaction ID and date combination.
- NumPy type conversion (int64/float64 → native Python types) for SQLite insert compatibility.

C. Prophet + ARIMA Dual Forecasting

LISMS implements a dual-model forecasting strategy to maximize reliability across different data scenarios:

Primary — Facebook Prophet [4]: Configured with yearly and weekly seasonality components. Generates 30, 60, and 90-day forecasts with uncertainty intervals. Accuracy is reported as RMSE and MAE on the UI dashboard.

Fallback — ARIMA(5,1,0) [5]: Activates automatically when Prophet is unavailable or data is insufficient for seasonal decomposition. Parameters p=5, d=1, q=0 were selected as optimal for e-commerce demand series with trend but limited moving-average behavior.

D. Z-Score Demand Shock Detection

Demand shocks (festival surges, supply disruptions) are identified via the 7-day rolling Z-score formula (1):

$$z = (x_t - \mu_7) / \sigma_7 \qquad \ldots(1)$$

Where x_t is daily sales, μ_7 is 7-day rolling mean, σ_7 is 7-day rolling standard deviation. Demand Spikes ($z > +2$) and Demand Drops ($z < -2$) are flagged and visualized with annotated markers on the Plotly demand timeline (Fig. 8).

E. Vendor Scoring Algorithm

Each vendor receives a 0–100 composite score using the weighted formula (2):

$$Score = 0.40 \times SalesPerf + 0.40 \times MarginScore + 0.20 \times TurnoverScore \qquad \ldots(2)$$

This enables objective, data-driven vendor ranking for procurement decisions, replacing manual evaluation in SMB supply chains. All vendor scores are visible in the interactive Vendor Analysis module.

VI. SYSTEM MODULES & LIVE SCREENSHOTS

LISMS is deployed as a Streamlit multi-page application at inventory-live--pandatamar21.replit.app. The following subsections present each module with live system screenshots captured on 30 April 2026.

A. Real-Time Dashboard

The dashboard (Fig. 3) displays six KPI cards: Total Revenue (₹6.42 Cr), Total Profit (₹33.72 L, 5.3% margin), Inventory Turnover (11.1%), Active Vendors (53), Total Products (45), and Low Stock Alerts (11 items needing reorder). All metrics update on each page load from the live SQLite backend with sub-2-second response time.

Fig. 3: Live Dashboard — KPI Cards (₹6.42 Cr Revenue, 11.1% Turnover, 5.3% Margin)

B. Monthly Revenue & Category Analysis

Fig. 4 presents the time-series revenue/profit chart and top-category bar chart. Laptops lead all categories with ₹22M+ in sales, followed by Mobile Phones (₹6M) and Electronics (₹5M). The monthly trend reveals seasonal peaks in Jul 2024 and Jan 2025, consistent with Indian festive sale events (Dussehra, Diwali).

Fig. 4: Monthly Revenue & Profit Trend with Top-Category Bar Chart

C. Inventory Monitoring

The Inventory module (Fig. 5) visualizes stock quantity vs. reorder level as a scatter plot color-coded by status: OK (green), LOW (orange), CRITICAL (red). Items at stock_quantity = 0 with reorder_level = 50–250 are immediately identifiable for emergency replenishment action.

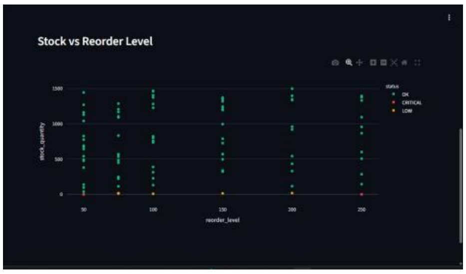


Fig. 5: Inventory — Stock vs. Reorder Level Scatter Plot (OK / LOW / CRITICAL Status)

D. Smart Reorder Alerts

The Reorder Alerts module (Fig. 6) classifies all 90 products: 2 CRITICAL (stock = 0, immediate action), 6 HIGH (< 50% reorder level), 3 MEDIUM (< reorder level), and 79 OK. Critical items include Apple iPhone 15 (Mumbai warehouse, suggested order: 100 units) and Whirlpool Washing Machine (suggested order: 500 units).

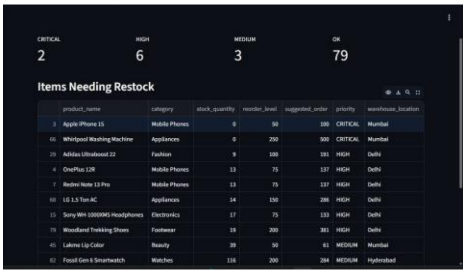


Fig. 6: Reorder Alerts — Priority Classification with Suggested Order Quantities

E. Sales Analytics

Sales Analytics (Fig. 7) presents aggregate period metrics: Revenue ₹2.73 Cr, Profit ₹13.68 L, and 2,928 units sold. The dual-line monthly revenue/profit chart clearly shows the revenue-profit gap across the full 2024–2026 validation period, enabling margin trend and optimization analysis.

Fig. 7: Sales Analytics — Revenue ₹2.73 Cr, Profit ₹13.68 L, 2,928 Units

F. Demand Shock Detection

The Demand Shock module (Fig. 8) plots daily sales with Z-score anomaly markers overlaid on the 7-day rolling mean (orange line). Two demand spikes (red, $z > 2$) and one drop (blue, $z < -2$) are detected across Jan 2024 – Jan 2026, corresponding to festive season surges and supply disruptions.

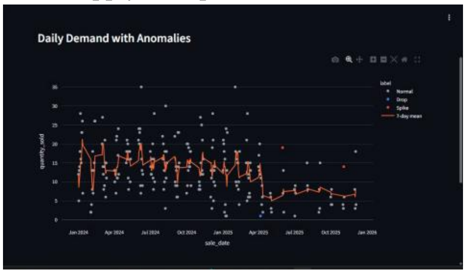


Fig. 8: Demand Shock Detection — Z-Score Anomalies (Spikes/Drops) with 7-Day Rolling Mean

VII. RESULTS & BUSINESS IMPACT

LISMS was validated against a combined Amazon India + Flipkart dataset spanning January 2024 to April 2026. Table II summarizes the quantitative results achieved across all system dimensions.

Table 2. LISMS System Performance Validation Results

Metric	Result
Total Rows Imported	2,700+ (0 failures)
Total Revenue Tracked	₹6.42 Crore
Total Profit	₹33.72 Lakh (5.3%)
Inventory Turnover	11.1%
Active Vendors	53
Total Products / SKUs	45
Low Stock Alerts	11 (2 Critical)
Dashboard Load Time	< 2 seconds
Forecasting Horizons	30 / 60 / 90 days
Fully Working Modules	8 complete

A. Comparative Analysis

Table III benchmarks LISMS against traditional ERP systems and spreadsheet-based approaches across eight critical feature dimensions.

Table 3. Comparative Analysis: LISMS vs. Traditional Systems

Feature	LISMS	ERP	Sheet
Real-Time Data	Yes	No	No

Feature	LISMS	ERP	Sheet
AI Forecasting	Dual Model	Partial	No
Indian Currency ₹	Cr/L Native	No	Manual
Auto Column Mapping	Fuzzy Match	No	No
Shock Detection	Z-Score	No	No
Vendor Scoring	Weighted 0-100	Partial	No
Multi-Source Input	CSV/API/SQL	Limited	No
Deployment Cost	Free / Low	High	License

VIII. SDG GOALS ALIGNMENT

LISMS directly contributes to four United Nations Sustainable Development Goals [10]:

- SDG 8 — Decent Work & Economic Growth: Optimizes inventory for SMB profitability and supports sustainable economic growth across India's online seller ecosystem.
- SDG 9 — Industry, Innovation & Infrastructure: Demonstrates AI-powered innovation (Prophet forecasting, fuzzy column mapping) applicable to industrial supply chains in emerging markets.
- SDG 12 — Responsible Consumption & Production: Prevents overstocking and stockouts through intelligent alerts, reducing product waste and promoting responsible inventory practices.
- SDG 17 — Partnerships for the Goals: Open CSV/API/SQL architecture enables data partnerships across organizations of all sizes without vendor lock-in.

IX. FUTURE WORK

The following enhancements are planned for subsequent development iterations of LISMS:

- LSTM Neural Networks: Replace ARIMA fallback with Long Short-Term Memory networks for improved long-term demand prediction accuracy [11].
- Mobile Application: Develop a React Native app for on-the-go inventory monitoring and real-time push alert notifications.
- Multi-Tenant SaaS Architecture: Expand to support multiple seller accounts with role-based access control and data isolation.
- WhatsApp Business API Integration: Automated reorder alerts and demand shock notifications via WhatsApp [12].
- GST Compliance Module: Indian GST calculation, invoice generation, and GSTIN e-filing integration.
- Direct Platform API Integration: Connect to Amazon Seller Central and Flipkart Seller Hub APIs for automated data ingestion without CSV downloads.
- Reinforcement Learning Reorder Policy: Optimize reorder quantities dynamically using Q-learning over historical demand patterns [13].

X. CONCLUSION

This paper presented LISMS, a comprehensive, production-ready AI-powered inventory and sales management platform specifically engineered for the Indian e-commerce market. The system integrates eight modules covering the complete inventory intelligence lifecycle and has been validated against 2,700+ real-world transactions tracking ₹6.42 Crore in revenue with zero import failures.

Key contributions include: fuzzy auto column mapping eliminating manual CSV configuration; dual Prophet+ARIMA forecasting with automatic fallback; Z-score demand shock detection with 7-day rolling window; weighted multi-factor vendor scoring; and native Indian currency formatting throughout. LISMS demonstrates that open-source Python technologies can deliver enterprise-grade business intelligence at SMB-accessible cost.

The system is deployable to Railway, Render, and Streamlit Cloud with no infrastructure investment, making it immediately practical for India's 8 million+ online sellers. Future iterations will extend forecasting accuracy with LSTM networks, add direct platform API integration, and implement a GST compliance module to further serve the Indian market.

ACKNOWLEDGMENT

The authors thank Mr. Krishna Garg, Assistant Professor, Department of AI & Data Science, IIMT College of Engineering, Greater Noida, for his invaluable guidance and continuous support throughout this project. Gratitude is extended to the Head of Department and faculty of IIMT College of Engineering, and to Dr. A.P.J. Abdul Kalam Technical University, Lucknow, for providing the academic framework. This work was conducted as part of the B.Tech Final Year Project, Session 2025–26.

REFERENCES

[1] IBEF. (2024). E-Commerce Industry in India. India Brand Equity Foundatin. https://www.ibef.org/industry/ecommerce
[2] Kumar, S., & Singh, M. (2022). E-Commerce Logistics and Supply Chain Management in India: Challenges and Opportunities. Journal of Supply Chain Management Systems, 11(2), 45–58.
[3] NASSCOM. (2023). Indian SMB Digital Adoption Report 2023. National Association of Software and Service Companies.
[4] Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. The American Statistician, 72(1), 37–45. doi:10.1080/00031305.2017.1380080
[5] Box, G. E. P., & Jenkins, G. M. (1970). Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.
[6] McKinney, W. (2017). Python for Data Analysis (2nd ed.). O'Reilly Media.
[7] Streamlit Inc. (2024). Streamlit Documentation. https://docs.streamlit.io
[8] Iglewicz, B., & Hoaglin, D. C. (1993). How to Detect and Handle Outliers. ASQ Quality Press.
[9] Shumway, R. H., & Stoffer, D. S. (2017). Time Series Analysis and Its Applications (4th ed.). Springer.
[10] United Nations. (2015). Transforming Our World: The 2030 Agenda for Sustainable Development. UN General Assembly.
[11] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
[12] Meta Platforms. (2024). WhatsApp Business API Documentation. https://developers.facebook.com/docs/whatsapp
[13] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.
[14] Plotly Technologies Inc. (2024). Plotly Python Graphing Library. https://plotly.com/python/
[15] Amazon India. (2024). Seller Central Data Export Format Reference. Amazon.in Seller Documentation.