Theory and Practice of Science and Technology
2025, VOL. 6, NO. 8, 58-65
DOI: 10.47297/taposatWSP2633-456911.20250608

Research on E-Commerce Product Inventory Monitoring and Sales Forecasting Based on the Spark Platform: A Case Study of Furniture Products

Chuqi Qiu

School of Management, Hangzhou Dianzi University, Hangzhou, Zhejiang, 31001, China

ABSTRACT
This study presents a forecasting and inventory management framework for e-commerce, integrating Long Short-Term Memory (LSTM) networks and Extreme Gradient Boosting (XGBoost) within a distributed computing environment using Apache Spark. The approach is tested on the UK Online Retail dataset for a specific furniture SKU. Key contributions include integrating deep learning and ensemble learning, implementing a scalable Spark-based pipeline, and linking forecasts directly to inventory replenishment strategies. Experimental results show complementary strengths: LSTM excels at capturing long-term trends, while XGBoost responds quickly to short-term fluctuations. The framework demonstrates practical potential for forecast-driven inventory optimization.

KEYWORDS
E-commerce forecasting; Inventory management; Long Short-Term Memory (LSTM); Extreme gradient boosting (XGBoost); Distributed computing; Spark

1 Introduction

E-commerce has transformed the global retail landscape, evolving from a supplementary sales channel into one of the primary revenue drivers worldwide. According to industry forecasts, global retail e-commerce sales are expected to exceed USD 7 trillion in the coming years, with a compound annual growth rate above 10%. This rapid growth is driven by widespread internet access, smartphone penetration, enhanced payment systems, and a rising demand for convenience and personalization.

With expansion comes operational challenges. Accurate sales forecasting and scientifically designed inventory management are essential for improving efficiency, reducing costs, and enhancing customer satisfaction. Demand volatility—caused by shifting consumer behavior, competition, and complex promotions—means even short-term forecasting errors can lead to lost sales or excessive stock.

E-commerce datasets are often high-dimensional, non-stationary, and subject to abrupt fluctuations such as spikes during large-scale promotions (e.g., Singles' Day, Black Friday) or dips from external shocks. These characteristics make traditional forecasting approaches difficult to apply.

Over time, research in e-commerce forecasting has evolved alongside advances in computational power and data availability. Early work relied on statistical models such as exponential smoothing, ARIMA, and SARIMA—valued for simplicity and interpretability, but less effective with high variability and nonlinear patterns. Later, machine learning methods like Support Vector Regression, Random Forests, and Gradient Boosted Trees provided better flexibility in capturing complex relationships. More recently, deep learning architectures—including LSTM, GRU, and Transformer-based models—have excelled in sequential modeling, integrating statistical rigor with computational sophistication to address the dynamic and data-rich e-commerce environment.

Despite advances, three challenges remain:

(1) Scalability – Many models are not optimized for distributed data processing.

(2) Integration with Inventory Management – Few directly link forecasts to automated, adaptive replenishment.

(3) Interpretability – Deep learning models often operate as "black boxes."

To address these gaps, this study proposes a Spark-based forecasting and inventory monitoring system integrating LSTM and XGBoost to:

(1) Balance predictive accuracy with interpretability.

(2) Scale effectively in industrial environments.

(3) Provide actionable inventory replenishment decisions.

By combining distributed computing with advanced machine learning, the research aims to bridge the gap between academic forecasting models and the operational needs of real-world supply chains.

2 Literature review

2.1 International research progress

Sales forecasting in developed economies has a mature theoretical base. Traditional statistical methods like ARIMA and SARIMA are effective for datasets with strong trends and seasonality but less capable of handling nonlinear interactions. Machine learning models such as SVM and Random Forest improved accuracy but often ignored temporal dependencies. Recurrent architectures like LSTM and GRU addressed these issues, enabling long-term dependency modeling. Hybrid approaches combining ARIMA with LSTM and ensemble learning methods like XGBoost have further enhanced accuracy and interpretability.

2.2 Domestic research progress

In China, research started later but developed rapidly. Early work adapted ARIMA with localized features such as holiday effects. More recently, LSTM has been widely used with additional variables from e-commerce big data, including browsing behavior, promotion intensity, and weather. XGBoost is frequently used for feature selection. Inventory studies often focus on national shopping events like "Double 11," where forecasts guide safety stock increases and replenishment frequency adjustments.

2.3 Comparative analysis of international and domestic research

While international research emphasizes methodological integration and deep integration with inventory optimization, domestic research focuses on application-driven approaches tailored to market dynamics.

Table 1 Comparison of international and domestic research

Dimension	International Focus	Domestic Focus
Research Start Point	Earlier, mature framework	Later start, rapid growth
Methods	ARIMA, LSTM, XGBoost, hybrids	Modified ARIMA, LSTM, XGBoost with local adaptations
Feature Selection	External variables (macro, weather)	Localized features (holidays, promotions)
Inventory Management	Integrated with models, adaptive strategies	Preparation for high-volatility events
Technical Architecture	Large-scale parallel computing, cloud deployment	Big data platforms with mixed distributed/single-node

2.4 Research trends and directions

Emerging trends in forecasting and inventory research include:
(1) Hybrid Model Integration — Combining statistical, ML, and deep learning models.
(2) Feature Diversification — Using macroeconomic, weather, and social media data.
(3) Forecast-Decision Integration — Tight coupling between forecasts and business actions.
(4) Computational Optimization — Balancing distributed and high-performance single-node processing.

3 Methodology

This study integrates distributed data processing with advanced forecasting models to enable scalable, interpretable, and operationally relevant inventory monitoring. The methodology consists of:(1) data acquisition and preprocessing(2) feature engineering and time series construction(3) dataset partitioning and modeling strategy(4) model design and training(5) performance evaluation(6) inventory simulation with replenishment optimization.

3.1 Data acquisition and cleaning

Data Source — UK Online Retail dataset (Dec 2010 – Dec 2011), ~540k records. Key fields: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country. Focus SKU: 85123A.

Cleaning — Remove missing/invalid entries, negative quantities/prices, convert dates, filter target SKU. Final dataset: ~ 280k records.

3.2 Overview of feature engineering

Feature engineering improves model performance by leveraging temporal aggregation, lag features, and normalization.

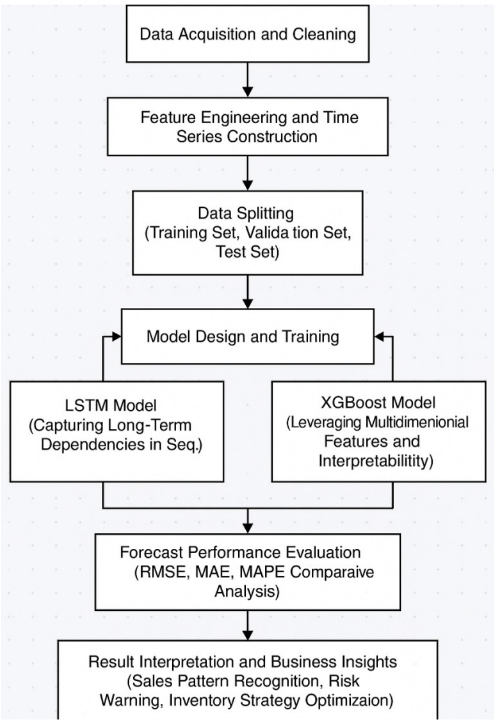

Figure 1 Workflow of forecasting methodology using LSTM and XGBoost

3.2.1 Daily aggregation

Daily sales are aggregated per date:

$$y_t = \sum_{i \in D_t} \text{Quantity}_i$$

where D_t is all transactions on day t. Temporal features include DayOfWeek and Month.

3.2.2 Lag features

Lagged variables capture historical dependencies:

$$\text{lag}_{k(t)} = y_{t-k} k \in \{1,7,14\}$$

These features capture short-term, weekly, and biweekly sales patterns.

3.2.3 Normalization

Standardization via Z-score:

$$x' = \frac{x - \mu}{\sigma}$$

ensures consistent scaling across datasets.

3.3 Dataset partitioning and modeling strategy

Chronological split:

(1) Training: before last 56 days.
(2) Validation: −56 to −29.
(3) Test: last 28 days.

3.4 Model design and training

The dual-model framework is particularly advantageous because it balances the LSTM's temporal sequence learning with XGBoost's strength in handling heterogeneous features. LSTM captures dependencies that may span weeks or months, while XGBoost rapidly adapts to short-term shifts and offers transparent feature importance rankings. This synergy ensures that the forecasting system is not only accurate but also interpretable, enabling decision-makers to trust and act upon the predictions.

3.4.1 Lstm model

The LSTM architecture is designed to capture long-range dependencies in sequential data while mitigating the vanishing gradient problem of traditional RNNs. Its internal mechanisms control information flow through three gates:

Forget Gate – determines what information from the previous cell state should be discarded:

$$f_t = \sigma\left(W_{f[h_{t-1}, x_t]} + b_f\right)$$

Input Gate and Candidate State – decides which new information should be stored in the cell state:

$$i_t = \sigma\left(W_{i[h_{t-1}, x_t]} + b_i\right)$$

$$\widetilde{C}_t = \tanh\left(W_{C[h_{t-1}, x_t]} + b_C\right)$$

Cell State Update – integrates retained old state and new candidate values:

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t$$

Output Gate – determines the output of the current step:

$$o_t = \sigma\left(W_{o[h_{t-1}, x_t]} + b_o\right)$$

$$h_t = o_t \odot \tanh\left(C_t\right)$$

By processing sequences of 14 consecutive days of historical features, the LSTM model can learn both short-term variations and long-term seasonal trends. The architecture comprises one LSTM layer with 64 units followed by a dense output layer. Adam optimizer (η=0.001) is used with Mean Squared Error (MSE) loss, and early stopping with patience of 10 epochs prevents overfitting.

3.4.2 XGBoost model

XGBoost constructs an ensemble of regression trees, each one built to minimize the residual errors of the previous trees. Its objective function is given by:

$$L = \sum_{t=1}^{n}\left(y_i - \widehat{y}_i\right)^2 + \sum_{k}\Omega(f_k)$$

where $\Omega\left(f_k\right)$ penalizes tree complexity to avoid overfitting.

The model uses calendar features (Day of Week, Month) and lag variables (Lag-1, Lag-7, Lag-14) to capture both temporal and historical dependencies.

Hyperparameters:

·$n_{estimators}$=500— Number of boosting rounds.

· max_depth=6— Controls model complexity and interaction depth.

· η=0.05— Learning rate, balancing step size and convergence speed.

· subsample=0.9— Row sampling rate to reduce variance.

· colsample_bytree=0.9— Feature sampling rate to increase diversity among trees.

By integrating the LSTM's ability to detect evolving temporal patterns with XGBoost's feature interpretability and computational efficiency, the framework ensures both high predictive accuracy and actionable insights for inventory management.

3.5 Evaluation metrics

3.5.1 Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\widehat{y}_t - y_t\right)^2}$$

3.5.2 Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|\widehat{y}_t - y_t\right|$$

3.5.3 Mean Absolute Percentage Error (MAPE)

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{\widehat{y_t} - y_t}{y_t} \right|$$

3.5.4 Symmetric MAPE (SMAPE)

$$\mathrm{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{2|\widehat{y_t} - y_t|}{|\widehat{y_t}| + |y_t|}$$

3.6 Inventory simulation and replenishment strategy

Simulation rule:

$$\mathrm{Stock}_{t+1} = \mathrm{Stock}_t - \mathrm{Demand}_t + Q_t$$

Replenishment trigger:

$$Q_t = f(x) = \begin{cases} 300, & \text{if } \mathrm{Stock}_t < 100 \\ 0, & \text{otherwise} \end{cases}$$

Initial stock: 500 units; safety stock: 100 units; horizon: 30 days.

4 Experimental results and analysis

This section presents the experimental findings derived from the proposed forecasting framework. Both LSTM and XGBoost models were trained and evaluated under identical preprocessing and feature engineering conditions. Performance was assessed using RMSE, MAE, MAPE, and SMAPE.

4.1 Overall model performance comparison

4.1.1 Background

Objective: Compare LSTM and XGBoost predictive performance for SKU 85123A over a 28-day test period, ensuring fair comparison through identical data splits and preprocessing.

4.1.2 Quantitative results

Table 2 Comparative performance of LSTM and XGBoost

Model	RMSE	MAE	MAPE (%)	SMAPE (%)
LSTM	176.643	136.416	87.03	157.04
XGBoost	115.948	98.244	393.12	68.73

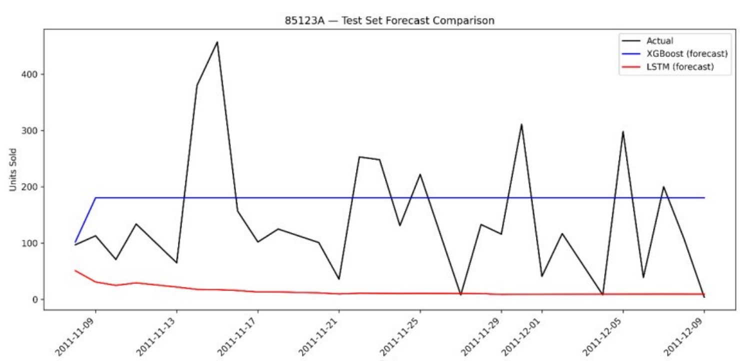

Figure 2 Forecast curves comparison for SKU 85123A over the test set

Key Findings:

XGBoost achieves lower RMSE, MAE, and SMAPE, suggesting better absolute error performance.

LSTM achieves lower MAPE, indicating higher proportional accuracy in low-sales cases.

High MAPE in XGBoost is due to percentage inflation when actual sales are near zero.

4.1.3 Business insight

In practice, this means that operational strategies can be tailored: XGBoost outputs can guide inventory policies for products with stable demand patterns, ensuring efficiency and minimizing overstock, while LSTM forecasts can serve as early warning indicators for sudden surges, allowing rapid response and flexible resource allocation.

4.2 Residual distribution analysis

4.2.1 BackgroundResiduals are defined as

$$\epsilon_t = y_t - \widehat{y_t}$$

Where y_t is actual sales, $\widehat{y_t}$ is predicted sales.Residual analysis identifies systematic bias:

Persistent negative residuals → underestimation risk

Persistent positive residuals → overestimation risk

4.2.2 Results and Interpretation

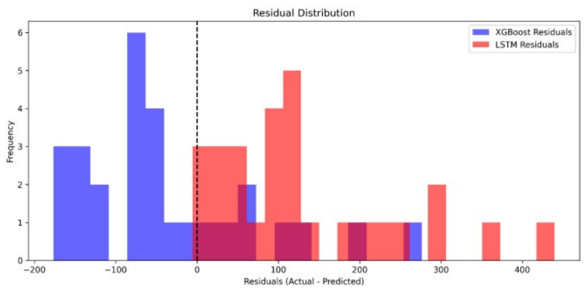

Figure 3 Residual distribution for LSTM and XGBoost on the test set

XGBoost residuals: Concentrated in negative range (approx. −150 to −50) → consistent underestimation.

LSTM residuals: Wider range (approx. −50 to +450) → frequent overestimation during some periods.

XGBoost is more stable but risks stockouts; LSTM reacts more to upward trends but risks overstocking.

4.2.3 Business insight

For organizations managing diverse product portfolios, such bias tendencies imply that a hybrid monitoring approach could be beneficial. By continuously comparing and contrasting outputs from both models, managers can dynamically adjust replenishment decisions, using XGBoost's conservative estimates for low-risk periods and LSTM's more aggressive forecasts for high-demand windows.

4.3 Feature importance analysis

4.3.1 Background

XGBoost feature importance (Gain) reveals the relative contribution of predictors.

4.3.2 Results

Month is most influential → strong seasonal cycles.

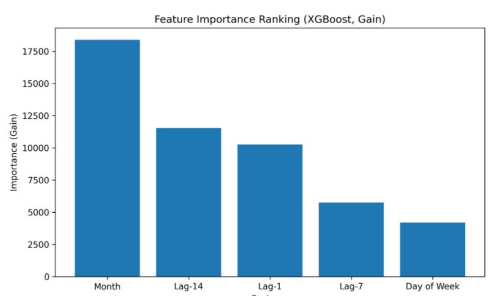

Figure 4 Feature importance ranking in XGBoost model

Lag-14 and Lag-1 follow → biweekly and previous-day sales important.
DayOfWeek least important → intra-week variation minimal.

4.3.3 Business insight

Beyond operational adjustments, feature importance analysis can also inform marketing strategies. For instance, identifying month as the dominant driver of demand could influence when promotional campaigns are scheduled, aligning marketing spend with historically high-demand periods.

4.4 Inventory simulation based on forecast results

4.4.1 BackgroundA

30-day inventory simulation using XGBoost forecasts tests replenishment strategy. Parameters: initial stock=500, safety stock=100, replenishment=300 units when threshold is breached.

4.4.2 Results

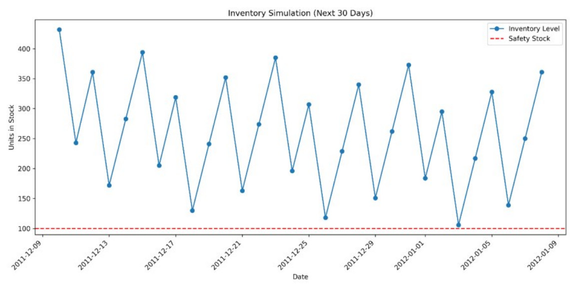

Figure 5 Simulated inventory levels over 30 days based on XGBoost forecasts

Inventory levels exhibit periodic replenishment.
Safety stock breaches trigger timely restocking.
Peak inventory levels remain stable post-replenishment.

4.4.3 Business insight

In a broader operational context, replenishment strategies can be further enhanced by integrating supplier lead times, transportation delays, and warehouse capacity constraints into the model. This would enable the transition from static to adaptive replenishment systems, improving both cost efficiency and service levels.5. Conclusions and Future Work

5 Conclusions and future work

5.1 Research conclusions

This study developed and validated a sales forecasting and inventory monitoring framework based on Apache Spark, integrating LSTM and XGBoost. Using the UK Online Retail dataset (furniture SKU 85123A), the research covered data preprocessing, feature engineering, model training, evaluation, and inventory simulation.

Key conclusions:

5.1.1 Model performance and applicability

LSTM captured long-term demand trends and seasonal patterns.
XGBoost responded faster to short-term fluctuations and offered higher interpretability.

5.1.2 Scenario-specific strengths

LSTM performed better in stable demand or seasonal peaks.
XGBoost was more effective in high-volatility promotional events.

5.1.3 Inventory strategy optimization

Forecast-informed replenishment reduced stockouts and improved turnover.

5.1.4 Distributed computing benefits

Spark accelerated large-scale processing, while Pandas allowed flexible small-scale analysis.

5.2 Research contributions

(1) Integration of LSTM and XGBoost for a balance between long-term accuracy and short-term responsiveness.
(2) End-to-end Spark Implementation for scalability in industrial applications.
(3) Forecast–Inventory Linkage that bridges predictive analytics and operational decision-making.
(4) Hybrid Analytical Architecture combining distributed and in-memory processing.

5.3 Research limitations

(1) Single-Product Focus — Results may not generalize to multi-SKU forecasting.
(2) Exclusion of External Variables — No integration of weather, macroeconomic data, or competitor information.
(3) Simplified Model Integration — No advanced model fusion (e.g., stacking, probabilistic blending).

5.4 Future work

(1) Cross-Category and Multi-Region Forecasting with transfer or multi-task learning.
(2) Integration of Multi-Source Data such as social media sentiment and weather.
(3) Advanced Architectures including Transformer-based time series models.
(4) Real-Time Forecasting with streaming platforms for dynamic safety stock adjustments.
(5) Closed-Loop Decision Systems for automated supply chain optimization.

5.5 Final remarks

Combining distributed computing and advanced ML models can enhance forecasting accuracy and operational efficiency in e-commerce. By linking predictions directly to inventory strategies, the proposed framework moves toward prescriptive analytics, laying the foundation for intelligent, automated supply chains.

References

[1] Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. International Journal of Forecasting, 37(2), 587–603.

[2] Wang, M., Liu, Y., Li, G., et al. (2024). Unlocking your sales insights: Advanced XGBoost forecasting models for Amazon products. arXiv preprint arXiv:2411.00460.

[3] Bandara, K., Shi, P., Bergmeir, C., et al. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In International Conference on Neural Information Processing (pp. 462–474). Cham: Springer.

[4] Salinas, D., Flunkert, V., & Gasthaus, J. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36(3), 1181–1191.

[5] Mukherjee, S., Shankar, D., Ghosh, A., et al. (2018). AR-MDN: Associative and recurrent mixture density networks for eRetail demand forecasting. Proceedings of the VLDB Endowment, 11(5), 553–565.

[6] Han, Y. J., & Gao, X. (2022). E-commerce product sales forecasting based on a combined machine learning model. Computer Systems Applications, 31(1), 315–321.

[7] Hu, B. W., & Li, J. (2021). E-commerce product sales forecasting based on multi-layer LSTM. Computer Science and Applications, 11(12), 3081–3090.

[8] Hu, Y. Y., Zhang, C., & Zheng, M. (2020). Design and implementation of a sales forecasting system based on LSTM. Computer and Network, 46 (18), 45–49.

[9] Jiang, W. W. (2019). Research on commodity sales forecasting based on WaveNet-LSTM network (Master's thesis). Guangdong University of Technology.

[10] Li, Z. H., & Zhang, K. L. (2020). Comparative study of automobile sales forecasting models based on BP algorithm and LSTM algorithm. Economic Research Guide, (20), 84–88.