

An Adaptive Hybrid ARIMA-XGBoost Framework with Real-Time Streaming for Retail Demand Forecasting

G Kavya Sri^{1*} & S Chandra Sekhar²

¹PG Scholar, Department of Computer Science and Engineering, UCEK, JNTUK, Kakinada, Andhra Pradesh, India

²Assistant Professor, Department of Computer Science and Engineering, UCEK, JNTUK, Kakinada, Andhra Pradesh, India

Abstract

Forecasting retail demand is a difficult undertaking since customer demand is influenced by temporal dependencies, changing sales patterns, nonlinear relationships, and differences between products and stores. This study introduces an adaptive hybrid forecasting approach that integrates Autoregressive Integrated Moving Average (ARIMA) with Extreme Gradient Boosting (XGBoost) in order to predict retail demand. ARIMA is employed to model the temporal structure of each individual demand series, and XGBoost is used to learn the residual patterns that are left after the statistical forecast has been carried out. An adaptive rolling forecasting procedure is included so that recent observations can be utilized to update the later predictions. The framework is assessed using the M5 retail forecasting dataset, keeping the demand data at the individual store-product level rather than aggregating the series. In order to apply the forecasting process in a real-time environment, Apache Kafka is used for continuous data ingestion and processing, and a Streamlit dashboard is provided to visualize the predictions, the errors, and the forecasting performance. The experimental results indicate an RMSE of 1.9796, a MAE of 0.9799, and a WAPE of 72.89%, which shows an 8.73% reduction in RMSE compared to the ARIMA baseline forecast. The results prove that combining statistical forecasting with machine learning-based residual correction and streaming infrastructure offers a practical solution for responsive retail demand forecasting.

Keywords: Retail demand forecasting, ARIMA, XGBoost, Hybrid forecasting, Real-time streaming, Apache Kafka, Machine learning

1. Introduction

Accurate retail demand forecasting is important for inventory planning, replenishment, and supply-chain decisions. However, retail demand varies across products and stores and often contains temporal dependencies, changing trends, and nonlinear patterns. Statistical methods such as Autoregressive Integrated Moving Average (ARIMA) can capture temporal relationships effectively, while machine learning models such as XGBoost can learn nonlinear

*Corresponding Author Email: kavyasrigubbala48@gmail.com

Published: 29 September 2026

DOI: <https://doi.org/10.70558/IJST.2026.v3.i3.241338>

Copyright © 2026 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

patterns from historical and engineered features. A single model may therefore fail to represent all characteristics of retail demand effectively.

This study proposes an adaptive hybrid ARIMA–XGBoost framework in which ARIMA generates the baseline forecast and XGBoost models the residual errors to provide nonlinear correction. A rolling forecasting mechanism incorporates newly available observations, while Apache Kafka enables continuous streaming of demand data and a Streamlit dashboard provides monitoring of predictions and forecasting performance. The framework is evaluated using the M5 retail dataset at the individual store-product level without aggregation. The proposed system combines forecasting, streaming, and visualization within a single framework for responsive retail demand prediction.

2. Review of Literature / Background

For many years, retail forecasting has been carried out with classical time-series models, chiefly ARIMA together with its seasonal and extended forms, because such models describe how successive demand values relate to one another and how underlying trends evolve. However, retail sales frequently contain nonlinear relationships, intermittent demand, promotional effects, and product- or store-specific variations that can limit the performance of conventional statistical models. Machine learning methods have therefore gained increasing attention in retail forecasting. Boosted decision-tree methods, XGBoost foremost among them, have proved especially effective for sales prediction because they infer

complex, non-additive dependencies directly from past sales figures and supporting contextual variables [1], [2]. The M5 forecasting competition further demonstrated the importance of machine learning for large-scale hierarchical retail forecasting, while also showing that computationally efficient statistical methods can remain competitive at fine-grained product-store levels [3]. Subsequent studies using the M5 dataset have similarly highlighted the trade-off between model complexity, feature availability, and forecasting performance across different levels of retail demand [4].

Hybrid forecasting has emerged as another important direction because statistical and machine learning models can capture complementary characteristics of demand. Studies combining classical time-series models with machine learning or deep learning have investigated architectures in which statistical models capture linear or temporal components while learning-based models capture nonlinear patterns [5], [6]. For example, hybrid approaches involving ARIMA and LSTM have demonstrated the potential of combining statistical forecasting with nonlinear sequence learning [5], while comparisons between SARIMA and LSTM in retail supply-chain forecasting have shown that model performance can vary according to the characteristics of individual demand series [7]. More recent research continues to investigate hybrid and ensemble approaches for retail forecasting, including combinations of statistical models, gradient boosting, and deep learning [8], [9]. Although deep learning models can achieve strong forecasting performance, their computational and training requirements can make lightweight statistical–machine learning combinations attractive for continuously operating forecasting systems. Recent retail forecasting research has also emphasized leak-safe lag-based feature engineering and scalable machine learning models for product-level forecasting [10]. Despite these developments, comparatively limited work has integrated

residual-based statistical–machine learning forecasting, adaptive rolling prediction, and real-time streaming infrastructure within a single framework while preserving fine-grained retail demand series. This study addresses this gap through an adaptive ARIMA–XGBoost framework combined with rolling forecasting and Kafka-based real-time streaming.

3. Objectives of the Study

The main objective of this study is to develop an adaptive hybrid ARIMA–XGBoost framework for retail demand forecasting while preserving individual store-product demand patterns. The study also aims to integrate the forecasting model with Apache Kafka for real-time data processing and a Streamlit dashboard for monitoring predictions and forecasting performance.

The specific objectives are to:

1. Develop an ARIMA model to capture the temporal characteristics of retail demand.
2. Apply XGBoost to model residual patterns and improve the ARIMA forecast.
3. Implement adaptive rolling forecasting using newly available demand observations.
4. Evaluate the hybrid model using RMSE, MAE, and WAPE.
5. Connect the framework to Kafka so that demand records stream continuously, and to Streamlit so that results can be viewed as they arrive.

4. Research Methodology

Four elements are brought together in the proposed system: statistical forecasting, residual correction driven by machine learning, adaptive forecast updating, and real-time streaming. The M5 retail dataset is maintained at the individual store-product level without aggregation. After chronological preprocessing, ARIMA generates the baseline demand forecast, and XGBoost learns the residual patterns using the prepared forecasting features.

The forecasting process uses an adaptive rolling mechanism in which newly available observations are incorporated into subsequent forecasting cycles. For real-time operation, Apache Kafka receives incoming demand records through a producer and transfers them through a Kafka topic to the forecasting consumer. The consumer applies the ARIMA–XGBoost pipeline and records actual demand, predicted demand, and forecasting errors. A Streamlit dashboard displays the resulting predictions and performance measures. The forecasting performance is evaluated using RMSE, MAE, and WAPE by comparing predicted demand with the corresponding actual observations. The complete workflow is illustrated in Figure 1.

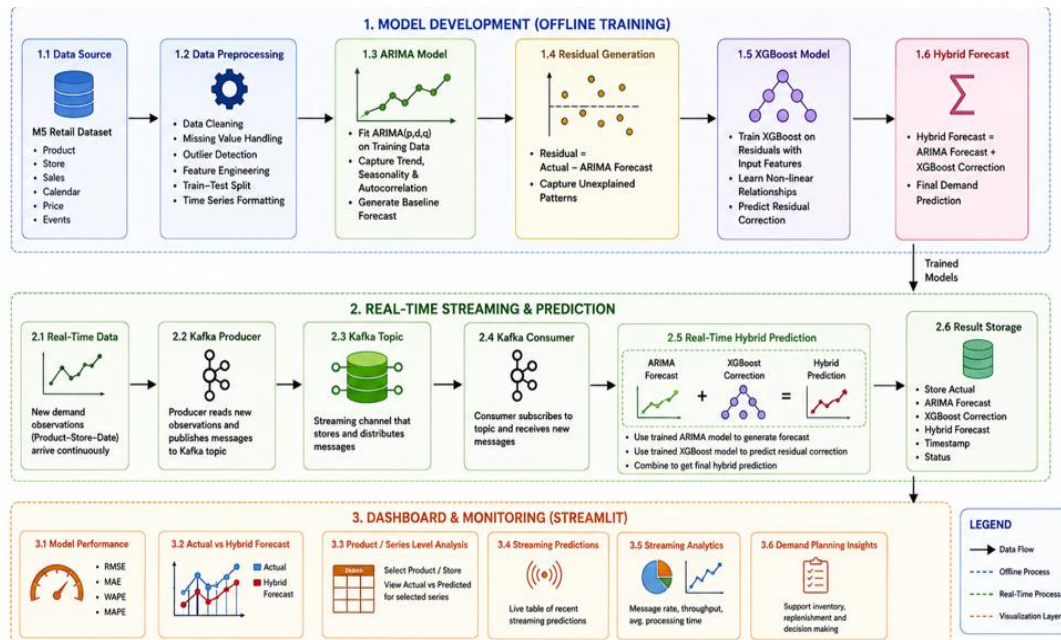


Fig1: Proposed System Architecture

A. Dataset and Data Preparation

The study uses the M5 retail forecasting dataset containing daily product-level sales information across Walmart stores. Demand observations are retained at the store-product level to preserve individual demand behaviour. The data are arranged chronologically, and relevant temporal, product, store, calendar, and demand-related features are prepared. Lagged and rolling statistical features are generated for the XGBoost model. Training and testing observations are separated chronologically to prevent future information from entering the model development process.

B. ARIMA-Based Demand Forecasting

ARIMA is used as the statistical forecasting component to model temporal dependencies in historical demand. It generates the initial forecast for the next observation, which serves as the baseline prediction for the hybrid framework.

$$y^{\text{ARIMA}} = \text{ARIMA}(y_1, y_2, \dots, y_{t-1})$$

The resulting ARIMA forecast is subsequently used to calculate the residual component for machine learning-based correction.

C. Residual Modeling Using XGBoost

Subtracting the ARIMA forecast for a given period from the demand actually observed in that period yields the residual:

$$e_t = y_t - y_t^{\text{ARIMA}}$$

These residuals represent demand patterns not adequately captured by ARIMA. XGBoost is trained using the prepared lag and forecasting features to learn nonlinear relationships within the residual series.

D. Hybrid Forecast Generation

The final demand prediction is generated by combining the ARIMA forecast with the XGBoost residual correction:

$$y_t^{\text{Hybrid}} = y_t^{\text{ARIMA}} + e_t^{\text{XGB}}$$

Thus, ARIMA is responsible for modeling the primary temporal behavior of demand, whereas XGBoost corrects the remaining nonlinear forecasting error. This complementary structure forms the core prediction mechanism of the proposed hybrid framework.

E. Adaptive Rolling Forecasting

An adaptive rolling forecasting mechanism is used to account for changes in demand over time. As new observations become available, the forecasting window is updated with recent demand information. These updated observations are then used for subsequent predictions, allowing the framework to respond to changing demand patterns rather than relying exclusively on a fixed historical window.

F. Real-Time Streaming Architecture

Apache Kafka is integrated to process incoming retail demand observations continuously. The producer sends demand records to a Kafka topic, and the consumer receives these records and applies the hybrid forecasting pipeline. The system records actual demand, predicted demand, and forecasting errors for monitoring. A Streamlit dashboard displays streaming predictions and forecasting metrics, connecting data ingestion, forecasting, evaluation, and visualization in a continuous workflow.

5. Results and Discussion

A. Forecasting Performance

The proposed ARIMA–XGBoost framework was evaluated using the M5 retail demand data. The overall forecasting performance is presented in Table 1. For the hybrid model the error measures were 1.9796 (RMSE), 0.9799 (MAE) and 72.89% (WAPE), with the RMSE figure representing an 8.73% gain on the ARIMA reference forecast.

Table 1: Overall Hybrid Model Performance

Metric	Value
RMSE	1.9796
MAE	0.9799
WAPE	72.89%
RMSE Improvement	8.73%

B. Real-Time Streaming Output

The trained forecasting pipeline was connected to Apache Kafka for processing incoming demand observations. For the displayed observation, the ARIMA forecast was 1, the XGBoost

residual correction was 0.003, and the resulting hybrid prediction was 1.00. Across 1,817 predictions drawn from five active M5 series, the streaming dashboard reported 1.6203 (RMSE), 0.9245 (MAE) and 64.17% (WAPE).

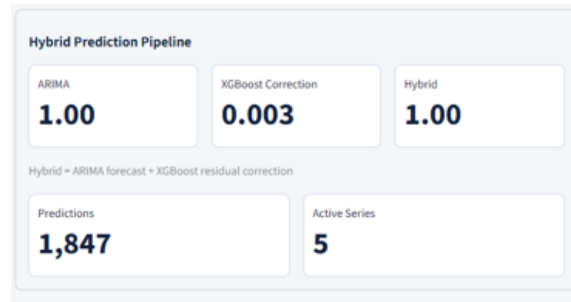


Fig 2: Real-Time Hybrid Prediction Output

The streaming analytics presented in Figure 3. The system processed 1,817 predictions across five active M5 series, demonstrating the operation of the forecasting model within the Kafka streaming environment.



Fig 3: Streaming Analytics

C. Prediction Analysis

The actual demand and hybrid prediction are compared in Fig 4. The prediction follows the general pattern of the observed demand, while the differences between the two series represent forecasting errors

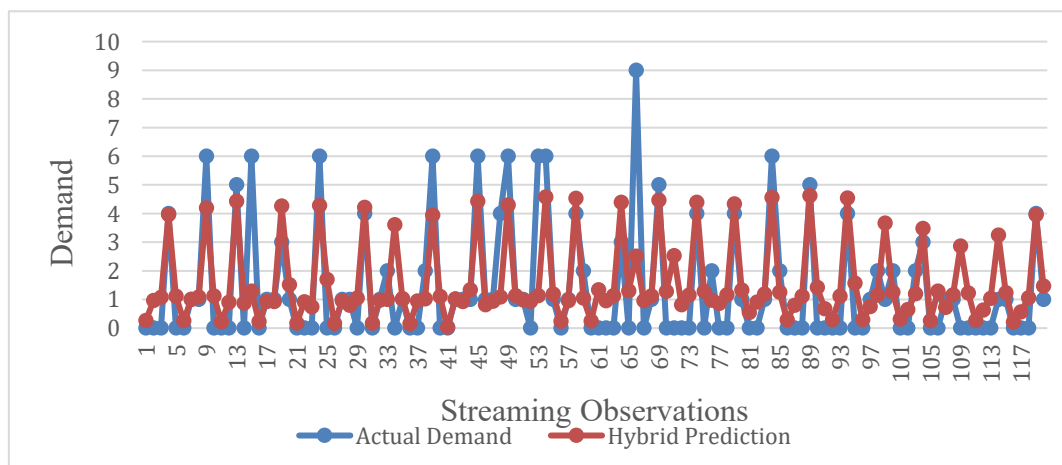


Fig 4: Actual Demand Vs Hybrid Prediction

The product-level comparison shown in Fig 5 presents actual and predicted demand for the active M5 product-store series. This visualization demonstrates that the system maintains product-store-level forecasting information during the streaming process.

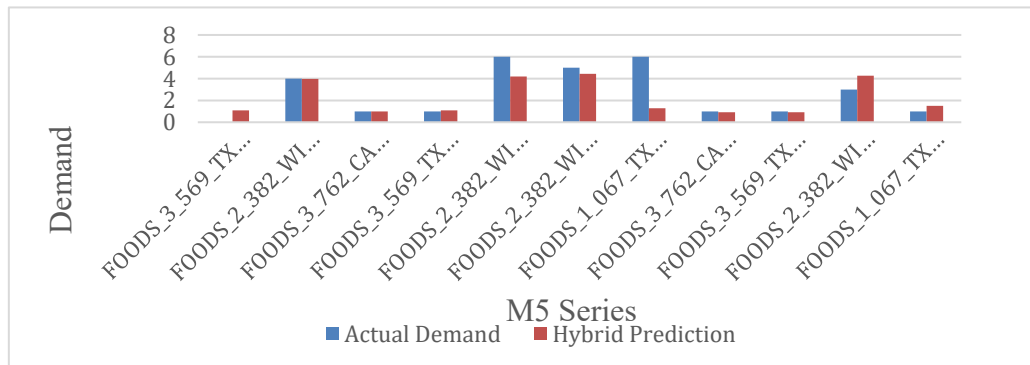


Fig 5: Actual Vs Predicted Demand by M5 Series

The variation in prediction error is presented in Fig 6. The graph provides a visual representation of forecasting deviations across the streaming observations and complements the numerical error measures.

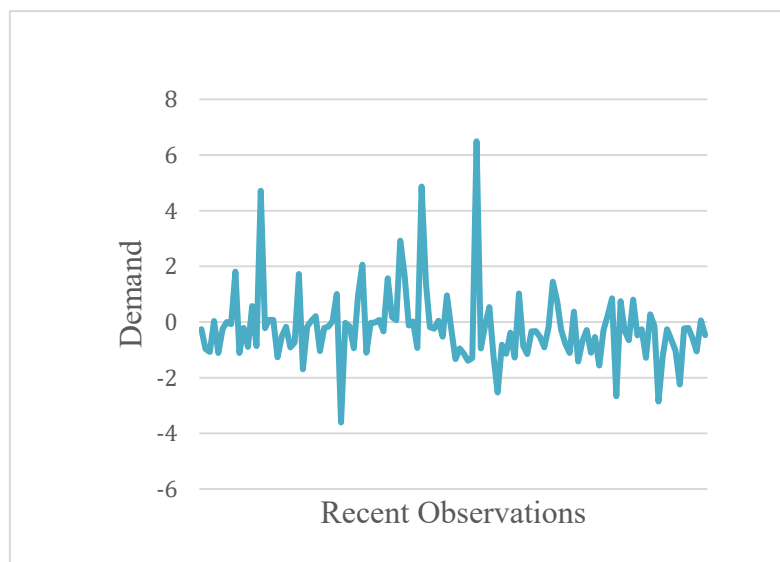


Fig 6: Live Forecast Error

D. Demand Planning Output

The final dashboard also provides a demand-planning signal based on the latest observation. As shown in Figure 7, the current demand is 1 unit, the forecast demand is 1.03 units, and the resulting forecast change is +2.9%, producing a Stable planning signal.

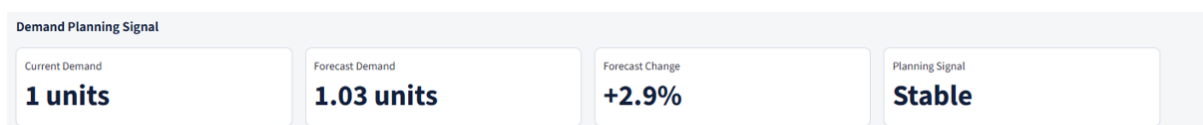


Fig 7: Demand Planning Signal

E. Discussion

The results show that the hybrid ARIMA–XGBoost framework improves the baseline forecasting process by combining ARIMA-based temporal modelling with XGBoost residual correction. The hybrid predictions generally follow the observed demand pattern, while the product-level analysis demonstrates that the system can retain and forecast individual M5 product-store series.

The integration with Apache Kafka enables the forecasting model to operate within a real-time streaming pipeline. The Streamlit dashboard brings the forecasting metrics, live predictions, prediction errors, and product-level comparisons together in a single interface, providing a practical view of both model behaviour and streaming activity.

G.Future Work

Future work can focus on improving the adaptability and scalability of the proposed forecasting framework. The model can be evaluated on larger retail datasets and additional product-store series to examine its performance across different demand patterns. Further improvements can include automated model parameter selection, additional demand-related features, and alternative machine learning methods for residual modelling.

The streaming component can also be extended to support larger data volumes and multiple forecasting services operating simultaneously. Future implementations can incorporate automated model updating, broader real-time monitoring, and additional decision-support features for inventory and replenishment planning. These extensions can help develop a more scalable and continuously adaptive retail forecasting system.

6. Conclusion

The proposed system presents a hybrid approach for retail demand forecasting by combining ARIMA for temporal demand modelling with XGBoost for residual correction. The framework retains product-store-level demand information from the M5 dataset and integrates the forecasting process with Apache Kafka to support real-time data processing.

The developed Streamlit dashboard provides a unified interface for monitoring forecasting performance, streaming predictions, prediction errors, and product-level demand behavior. Overall, the proposed framework demonstrates the feasibility of combining hybrid forecasting, real-time streaming, and visual analytics into a practical retail demand forecasting system.

References

- [1] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “The M5 competition: Background, organization, and implementation,” *International Journal of Forecasting*, vol. 38, no. 4, pp. 1325–1336, 2022, doi: 10.1016/j.ijforecast.2021.07.007
- [2] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “M5 accuracy competition: Results, findings, and conclusions,” *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022, doi: 10.1016/j.ijforecast.2021.11.013

- [3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785
 - [4] S. Smyl, “A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting,” *International Journal of Forecasting*, vol. 36, no. 1, pp. 75–85, 2020, doi: 10.1016/j.ijforecast.2019.03.017
 - [5] C. Bergmeir, R. J. Hyndman, and B. Koo, “A note on the validity of cross-validation for evaluating autoregressive time series prediction,” *Computational Statistics & Data Analysis*, vol. 120, pp. 70–83, 2018, doi: 10.1016/j.csda.2017.11.003
 - [6] G. Claeskens, J. R. Magnus, A. L. Vasnev, and W. Wang, “The forecast combination puzzle: A simple theoretical explanation,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 754–762, 2016, doi: 10.1016/j.ijforecast.2015.12.005.
 - [7] A. A. Syntetos, J. E. Boylan, and J. D. Croston, “On the categorization of demand patterns,” *Journal of the Operational Research Society*, vol. 56, no. 5, pp. 495–503 doi: 10.1057/palgrave.jors.2601841
 - [8] “Forecasting Sales in Retail with XGBoost and Iterated Multi-step Ahead Method,” in Proceedings of the 2022 IEEE 16th International Conference on Signal Processing and Information Technology, 2022, doi: 10.1109/SST55530.2022.9954658
 - [9] A. Choubey, “Hybrid Models for Retail Demand Forecasting: Integrating Classical Time-series and Machine Learning Approaches,” *Asian Journal of Research in Computer Science*, vol. 18, no. 8, pp. 113–123, 2025, doi: 10.9734/ajrcos/2025/v18i8744.
 - [10] “Improving Forecast Accuracy for Retail Supply Chains Using ARIMA and Machine Learning Approaches,” *Transportation Research Procedia*, vol. 96, pp. 524–531, 2026, doi: 10.1016/j.trpro.2026.04.024
- .