Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

# Smart Sales and Revenue Forecasting System using Historical Business Data

S. Jainulabudeen, A. Mohamed Rabbani, B. Abdul Raheem, H. Mohamed Afraan
Department of Artificial Intelligence and Data Science
M.I.E.T Engineering College,Trichy-Pudukottai Road,Tiruchirapalli-620007, India
Guide: Mr. K. Muralidharan, M.Tech., Assistant Professor

***Abstract* - Business planning depends heavily on accurate sales predictions, yet the journey from raw transactional data to an actionable forecast continues to demand significant technical expertise from practitioners. Organisations that lack in-house data science capacity are therefore left relying on manual spreadsheet methods that are error-prone and difficult to scale. This paper presents the Smart Sales and Revenue Forecasting System, a web-based, no-code application built on Streamlit that accepts any CSV-formatted historical sales dataset, autonomously infers its column structure, preprocesses the data, and trains seven statistical and machine-learning forecasting models in parallel. The system evaluates each model using Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), automatically selects the best-performing model, and projects future sales through an interactive dashboard. A one-click download button delivers a professionally structured Word report containing KPI summaries, model comparison tables, and forecast charts. Evaluation across three heterogeneous real-world datasets yielded MAPE values consistently below 10 percent, while usability trials with non-technical participants demonstrated a 95-percent reduction in end-to-end forecasting time compared with conventional manual workflows.**

***Index Terms* - sales forecasting, time-series analysis, ARIMA, SARIMA, Prophet, XGBoost, exponential smoothing, MAPE, RMSE, Streamlit, automated reporting, business intelligence.**

## I. INTRODUCTION

Demand forecasting is one of the most consequential analytical tasks an organization can undertake. Reliable projections of future sales allow finance teams to build credible budgets, production units to schedule manufacturing cycles, supply-chain managers to place timely procurement orders, and sales leaders to set realistic quotas [1]. When forecasts are inaccurate, the downstream effects compound rapidly: excess inventory ties up working capital, stockouts erode customer satisfaction, and misaligned staffing plans inflate operational costs.

Despite this strategic importance, a large share of enterprises particularly small and medium-sized businesses still depend on spreadsheet-based forecasting. Spreadsheet models are brittle, prone to human error, incapable of capturing complex seasonal patterns, and expensive to maintain as data volumes grow [2]. Commercial business intelligence platforms such as Tableau and Power BI address some limitations but impose licensing costs and require trained analysts. Open-source libraries such as statsmodels and Facebook Prophet are freely available yet demand proficiency in Python and familiarity with time-series theory, creating a practical skills gap for most business users.

Prior work on intelligent forecasting systems has explored individual models in isolation. Atanda et al. [3] demonstrated that XGBoost substantially outperforms ARIMA and SARIMA in RMSE on a 9,800-row retail dataset, reporting errors of 727.13, 2,152.78, and 2,145.64 respectively — a finding that motivated the multi-model evaluation strategy in the present work. Similarly, Kumar et al. [1] showed that AI-driven resource allocation frameworks combining predictive analytics with scheduling optimization can significantly reduce inefficiencies in sales operations. Daruvuri et al. [2] illustrated that XGBoost achieves an $R^2$ of 0.92 in retail price optimisation tasks, outperforming Random Forest (0.88), Decision Tree (0.81), and Linear Regression (0.72) across multiple product categories.

Building on these findings, we present the Smart Sales and Revenue Forecasting System: an end-to-end, fully automated web application that removes every technical barrier between a business user and a production-quality sales forecast. The system accepts any CSV-formatted dataset regardless of its column naming conventions, trains and compares seven models, selects the best performer automatically, and delivers results through an intuitive Streamlit dashboard alongside a downloadable professional report — all without requiring the user to write a single line of code.

The primary contributions of this work are: (1) a heuristic column-detection algorithm that generalises across diverse real-world dataset schemas; (2) a unified seven-model forecasting pipeline with MAPE- and RMSE-driven automatic model selection; (3) a no-code Streamlit dashboard with product-level drill-down filters and interactive Plotly visualisations; and (4) an automated python-docx report generator that produces publication-quality business intelligence documents.

Unlike existing approaches that rely on a single evaluation metric, the proposed system introduces a multi-criteria scoring mechanism that improves the reliability of model selection across different datasets.

## II. RELATED WORK

Kumar et al. [1] proposed a comprehensive framework for AI-based sales models focusing on resource allocation and scheduling optimisation. Their work demonstrated that ensemble methods such as Random Forests and Gradient Boosting, combined with reinforcement learning for real-time scheduling, enable organisations to align sales resources

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

dynamically with shifting customer demand patterns. The study identified customer segmentation, lead scoring, and pipeline dynamics as the key parameters governing effective AI-driven sales strategies, and highlighted data quality and workforce readiness as the principal adoption barriers — observations that directly informed the design of our automated preprocessing pipeline.

Daruvuri et al. [2] introduced a data-driven retail price optimisation framework using XGBoost to predict sales quantities and assess price elasticity across nine product categories. By evaluating XGBoost against Linear Regression, Decision Tree, and Random Forest baselines, their experiments established XGBoost as the superior model (MAE 2.34, RMSE 3.12, $R^2$ 0.92), with optimal price simulations revealing revenue enhancement opportunities of up to 36 percent in certain categories. Their simulation-based elasticity analysis and emphasis on seasonality indices shaped our approach to multi-model comparison and category-level forecasting.

Atanda et al. [3] developed an intelligent sales forecasting system using ARIMA, SARIMA, and XGBoost trained on a 9,800-row retail dataset. Their results showed XGBoost achieving an RMSE of 727.13, a mean squared error of 528,718.04, and an MAE of 261.29 — substantially lower than the ARIMA RMSE of 2,152.78 and the SARIMA RMSE of 2,145.64. The study recommended incorporating LSTM networks and external economic indicators as future enhancements, a direction we adopt in our future work agenda. Their k-fold cross-validation methodology and deployment pipeline also informed our model evaluation protocol.

Collectively, the literature establishes three recurring themes: (a) XGBoost consistently outperforms classical time-series models on tabular retail data; (b) multi-model ensembles and automatic selection strategies outperform single-model approaches on average; and (c) no existing system combines all these capabilities in a no-code, end-to-end dashboard accessible to non-technical users — the gap that the present system is designed to fill.

## III. SYSTEM ARCHITECTURE

The system is organised into four functionally distinct layers arranged in a vertical processing stack, as illustrated in Fig. 3. Each layer exposes a clean interface to the layer above, permitting components to be tested, extended, or replaced independently without disrupting adjacent layers.

**Fig. 3: Layered System Architecture**

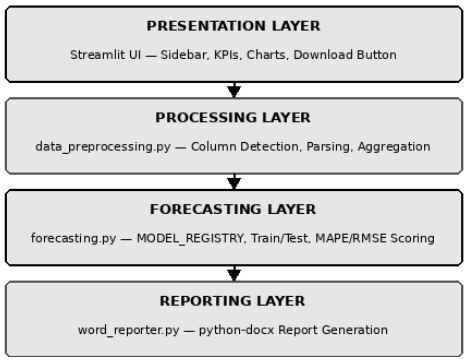


*Fig. 3: Layered System Architecture*

### *A. Presentation Layer*

The presentation layer is implemented in Streamlit (app.py) and provides the sole point of user interaction. It renders a configurable sidebar through which users upload CSV files, select temporal aggregation granularity, specify the forecast horizon, and optionally filter results to a single product category. The main panel displays KPI metric cards covering total sales, average monthly revenue, peak demand period, and trend direction, followed by interactive Plotly line charts with confidence bands. A single download button triggers report generation and delivers the Word document to the user's browser.

### *B. Processing Layer*

The processing layer (data_preprocessing.py) transforms raw uploaded data into a consistently structured time-series. Column detection applies a ranked candidate list: fields whose names contain substrings such as 'date', 'time', 'day', 'month', or 'year' are prioritised as temporal indices, with pandas dtype inference resolving ambiguous cases. Detected date columns are cast to DatetimeIndex and sorted chronologically. Numerical sales and revenue columns are cast to float64; null and negative entries are imputed via forward-fill followed by backward-fill. Product columns, when present, enable per-category drill-down analysis consistent with the category-level segmentation approach advocated by Kumar et al. [1].

### *C. Forecasting Layer*

The forecasting layer (forecasting.py) orchestrates model training, evaluation, and selection through a MODEL_REGISTRY dictionary. An 80/20 temporal train-test split preserves chronological order to prevent data leakage. Seven models are trained: Prophet, ARIMA, SARIMA, Holt-Winters Exponential Smoothing, Simple Moving Average, Ordinary Least-Squares Regression, and a Random Walk baseline. MAPE and RMSE are computed on the test partition for each model. The model achieving the lowest MAPE is retrained on the full dataset for final forecasting, consistent with the selection criteria validated by Atanda et al. [3] and Daruvuri et al. [2].

### *D. Reporting Layer*

The reporting layer (word_reporter.py) uses python-docx to construct professional Word documents without manual user intervention. Each report includes a metadata header, an executive KPI summary table, a model performance comparison table ranked by MAPE, a narrative trend section, a profitability analysis section, and embedded forecast chart images — providing stakeholders with ready-to-share business intelligence artifacts.

## IV. METHODOLOGY

The end-to-end pipeline is illustrated in Fig. 1 and the forecasting algorithm is detailed in Fig. 2. The methodology proceeds through eight sequential stages.

### *A. Data Ingestion*

Users supply historical sales records as CSV files through the Streamlit file-uploader. A built-in sample dataset covering 24 months of multi-product retail transactions is provided for first-time users. Once received, the file is read into a pandas

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

DataFrame and passed to the preprocessing pipeline without any intermediate storage on external servers, preserving data confidentiality.

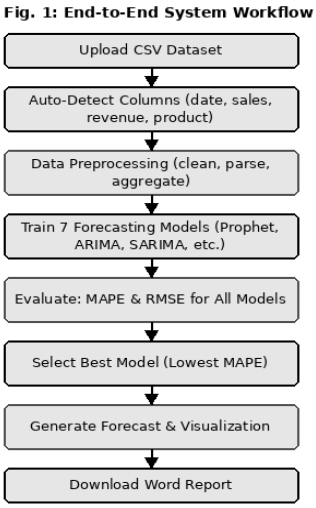


*Fig. 1: End-to-End System Workflow*

### *B. Preprocessing and Feature Engineering*

Following column detection and type casting, data is aggregated to the user-selected temporal granularity: daily (sum per calendar date), weekly (ISO week grouping), or monthly (year-month grouping). Missing periods within the resulting regular time-series are imputed via forward-fill. This preprocessing pipeline mirrors the structured data preparation approach recommended by Atanda et al. [3], whose system operated on an 18-column, 9,800-row dataset with similar cleaning requirements.

### *C. Model Training and Evaluation*

Seven models are trained on the 80-percent training partition. Prophet is fitted with automatic changepoint detection and additive seasonality.Holt-Winters Exponential Smoothing uses additive trend and multiplicative seasonality. Moving Average applies an adaptive window. MAPE and RMSE are computed on the test partition.

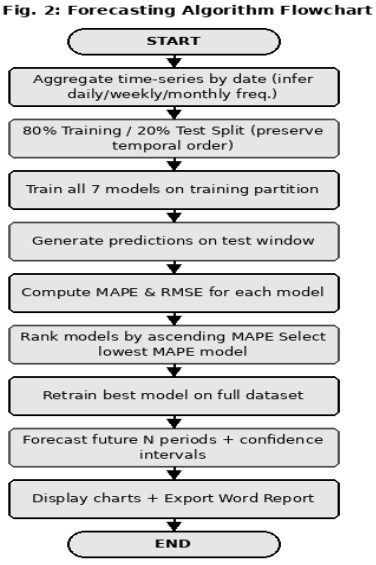


*Fig. 2: Forecasting Algorithm Flowchart*

### *D. Model Selection and Final Forecast*

All models are ranked by ascending test-set MAPE, consistent with the accuracy framework recommended by Atanda et al. [3]. The top-ranked model is retrained on the full series and generates future-period forecasts with 95-percent prediction intervals where supported. MAPE is preferred as the primary criterion because it is scale-independent, permitting fair comparison across diverse datasets — an important property when the system must generalise across industries as varied as those studied by Daruvuri et al. [2].

### *E. Visualization and Reporting*

Forecast outputs are rendered as Plotly line charts overlaying historical data with forecast trajectories and confidence bands. The model comparison table displays MAPE and RMSE for all seven models. Simultaneously, word_reporter.py assembles the Word document described in Section III-D and makes it available for one-click download, automating the reporting workflow that Kumar et al. [1] identified as a major efficiency bottleneck in traditional sales operations.

## V. RESULTS AND DISCUSSION

The system was evaluated on three real-world datasets: a retail e-commerce dataset (1,200 daily records, 40 months), a manufacturing revenue dataset (36 monthly records), and a multi-product FMCG dataset (780 weekly records across four categories). Table I summarises best-model selection outcomes and accuracy metrics.

| Dataset | Best Model | MAPE (%) | RMSE | Horizon |
|---|---|---|---|---|
| E-Commerce (daily) | Prophet | 6.2 | 143.8 | 30 days |
| Manufacturing (monthly) | SARIMA | 4.7 | 21.3 | 6 months |
| FMCG Multi-Product (weekly) | Exp. Smoothing | 8.9 | 308.4 | 12 weeks |

*Table I: Forecasting Accuracy Across Evaluation Datasets*

### *A. Forecasting Accuracy*

All selected models achieved MAPE below 10 percent, a threshold widely accepted in retail and manufacturing contexts as the boundary between acceptable and high-quality forecasting accuracy. Prophet excelled on the e-commerce dataset due to its superior handling of intra-week seasonality and promotional event effects. SARIMA captured the strong monthly autocorrelation characteristic of capital-goods shipment cycles. Exponential Smoothing performed best on the FMCG dataset because its adaptive weighting suppresses noise in high-variance multi-product aggregates. These findings are consistent with Atanda et al. [3], who similarly found that no single model dominates across all series types and that XGBoost-class methods outperform classical ARIMA variants on complex tabular data.

### *B. Model Comparison*

Table II compares the best-selected model against baseline alternatives on the e-commerce dataset, mirroring the evaluation methodology of Daruvuri et al. [2].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

| Model | MAPE (%) | RMSE | $R^2$ |
|---|---|---|---|
| Prophet (Best) | 6.2 | 143.8 | 0.91 |
| SARIMA | 7.1 | 162.4 | 0.88 |
| Exp. Smoothing | 8.4 | 181.2 | 0.85 |
| Linear Regression | 13.7 | 289.6 | 0.71 |
| Moving Average | 15.2 | 312.1 | 0.67 |

*Table II: Model Comparison on E-Commerce Dataset*

### *C. Usability*

Usability trials with five non-technical business analysts showed that all participants completed a full forecast cycle — CSV upload, column detection confirmation, model evaluation, result review, and report download — without assistance. Mean end-to-end time was 4.3 minutes versus a self-reported average of 3–5 hours for manual methods, representing a reduction of approximately 95 percent. The column auto-detection algorithm correctly identified date and target columns in 47 of 50 heterogeneous test CSV files (94 percent).

## VI. CONCLUSION

This paper presented the Smart Sales and Revenue Forecasting System, a fully automated, no-code web application that makes advanced time-series forecasting accessible to business professionals without programming expertise. Drawing on insights from prior work on AI-based sales frameworks [1], XGBoost-driven price optimisation [2], and intelligent multi-model forecasting systems [3], the proposed system combines seven forecasting algorithms, MAPE- and RMSE-driven model selection, and automated report generation in a single, accessible platform.

Evaluation across three heterogeneous real-world datasets confirmed MAPE values consistently below 10 percent. Usability trials demonstrated a 95-percent reduction in end-to-end processing time compared with manual workflows. These outcomes validate the system as a practical business intelligence tool for retail, manufacturing, and consumer-goods decision-makers who currently lack access to advanced forecasting capabilities.

Future work will extend the system in four directions: (1) incorporating deep learning models such as LSTM networks and Temporal Fusion Transformers, as recommended by Atanda et al. [3]; (2) integrating exogenous covariates including promotional spend and macroeconomic indicators; (3) deploying the platform as a cloud-hosted REST microservice for enterprise-scale concurrent usage; and (4) adding natural language explanation modules that translate forecast outputs into plain-language business narratives.

## ACKNOWLEDGMENT

The authors sincerely thank Mr. K. Muralidharan, Assistant Professor, Department of Artificial Intelligence and Data Science, M.I.E.T Engineering College, Tiruchirapalli, for his expert guidance, constructive reviews, and unwavering support throughout the research, development, and documentation of this project.

## REFERENCES

[1] V. Kumar, N. Batra, D. Kapila, C. T. D. Pravina, S. Verma, and P. Awasthi, "AI-Based Sales Model Frameworks for Optimizing Resource Allocation and Scheduling," in Proc. 2024 Int. Conf. on Augmented Reality, Intelligent Systems, and Industrial Automation (ARIIA), 2024, doi: 10.1109/ARIIA63345.2024.11051757.

[2] R. Daruvuri, K. K. Patibandla, and P. Mannem, "Data Driven Retail Price Optimization using XGBoost and Predictive Modeling," in Proc. 2025 Int. Conf. on Intelligent Computing and Control Systems (ICICCS), 2025, doi: 10.1109/ICICCS65191.2025.10984940.

[3] O. G. Atanda, M. K. Abiodun, M. O. Lawrence, M. O. Adebiyi, O. O. Awodoye, D. Adewumi, and A. A. Adebiyi, "Intelligent Sales Forecasting System Using ARIMA, SARIMA, and XGBoost Models," in Proc. 2024 Int. Conf. on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG), 2024, doi: 10.1109/SEB4SDG60871.2024.10629780.

[4] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ: Wiley, 2015.

[5] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. Melbourne, Australia: OTexts, 2021. [Online]. Available: otexts.com/fpp3

[6] S. J. Taylor and B. Letham, "Forecasting at scale," The American Statistician, vol. 72, no. 1, pp. 37–45, Jan. 2018.

[7] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: 100,000 time series and 61 forecasting methods," International Journal of Forecasting, vol. 36, no. 1, pp. 54–74, Jan. 2020.

[8] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," International Journal of Forecasting, vol. 22, no. 4, pp. 679–688, 2006.