Smart Supply Chain & Inventory Predictor using Time-Series Forecasting

Payal Ravi More[1]
morepayal5017@gmail.com
Student MCA-II

Ritika Ganpat Waghmare[2]
ritikaw27@gmail.com
Student MCA-II

Prof. Tejas Pawar[3]
tjspwr2@gmail.com
Assistant Professor- MCA

[1&2]Student, [3]Assistant Professor

Yashaswi's International Institute of Management Science, Pune (Savitribai Phule Pune University)

Abstract:Both stockouts and overstocking hurt e-commerce and retail businesses, and retail demand is shaped by seasonality, holidays and promotions that static or moving-average planning handles poorly. This study designs and evaluates a time-series demand forecasting and inventory decision framework. We compare a statistical model (ARIMA/SARIMAX), an additive model (Prophet) and a deep-learning model (LSTM) on public retail sales data using chronological splits, and convert forecasts into safety-stock and reorderpoint recommendations that are tested in an inventory simulation against a fixed-threshold policy. The pipeline is delivered through InventoryIQ, a web dashboard that shows actual versus predicted sales and reorder alerts.

Keywords:Demand forecasting, inventory optimisation, ARIMA, Prophet, LSTM, safety stock, supply chain analytics.

I. INTRODUCTION

A. Background

Keeping inventory at the right level is central to the profitability and customer satisfaction of retail and e-commerce firms. Too little stock loses sales and customer loyalty; too much ties up capital and raises warehousing cost and the risk of obsolescence or spoilage. The rapid growth of digital commerce in India and elsewhere has made this balance harder to hold, because product ranges are wide, demand changes quickly and customers expect fast delivery.

Retail sales series are difficult to plan for because they combine a long-term trend, several seasonal cycles (weekly, monthly, yearly), holiday spikes and promotion effects. Manual estimates and static reorder rules cannot represent these non-linear temporal patterns, and they react to shortages rather than anticipating them.

B. Motivation

Historical transaction data allows demand to be forecast with statistical and machine-learning models. Yet forecasting results are often reported as error tables, detached from the software that a store manager uses to place orders. A forecast only creates value when it is converted into a stocking decision, so the study evaluates forecasting accuracy and inventory outcomes together, and delivers both through a dashboard.

C. Objectives

- Analyse historical retail transactions to identify trend,seasonality and promotion effects.
- Build and compare ARIMA/SARIMAX, Prophet and LSTMforecasting models.
- Compare models using RMSE, MAPE and R^2 (with MAE andMASE as supporting measures).
- Compute reorder points and safety stock from forecasts andforecast error.
- Design and implement the InventoryIQ web dashboard onReact, a REST API and PostgreSQL.
- Visualise actual versus predicted sales for audit andpresentation.

D. Contributions

(1) A like-for-like comparison of ARIMA, Prophet and LSTM on retail demand using chronological evaluation. (2) A forecast-to-inventory step that turns predictions and forecast error into safety stock and reorder points, evaluated in simulation against a fixed-threshold policy. (3) InventoryIQ, a full-stack dashboard for demand insight and reorder alerts. We do not claim a new forecasting algorithm; all three models are established. *E. Organisation of the paper* Section II reviews related work. Section III formalises the problem and hypotheses. Section IV describes data and features, Section V the forecasting models, and Section VI the inventory model. Section VII presents the system design, Section VIII the experimental protocol, and Section IX the results. Sections X to XII cover discussion, limitations and conclusions.

II. RELATED WORK

A. Statistical forecasting

The ARIMA family [1] gives a sound basis for short-term linear dynamics, using autoregressive and moving-average terms on a differenced series. Seasonal extensions add period-specific terms. Such models are transparent and need little data, but they are limited when several seasonal patterns and non-linear holiday effects are present, as is common in e-commerce. Hyndman and Athanasopoulos [6] describe practical use of exponential smoothing and ARIMA, including the importance of simple benchmarks.

B. Additive decomposition models

Prophet [2] decomposes a series into trend, seasonality and holiday terms and tolerates missing data and trend changes. Analysts can add domain knowledge through holiday calendars and extra regressors, which explains its wide adoption in business forecasting.

C. Deep learning

Recurrent networks, and LSTM networks in particular [3], capture long-range dependencies and non-linear relations through gated memory cells. They can use many covariates and learn patterns across series, but they need more data, tuning effort and compute, and do not always beat simple baselines. Large forecasting competitions such as M5 [8], which used Walmart retail data, are a useful reference for how methods compare on retail demand.

D. Inventory control

Classical inventory theory sets reorder points and safety stock from lead-time demand and its variability for a target service level [10].

used. Studies often treat forecasting and the resulting stocking decision separately; here they are evaluated together.

E. Forecast evaluation

Scale-free and scaled error measures [9] and tests for comparing forecast accuracy [11] guide the metrics and tests used here. MAPE is popular but undefined at zero demand and biased towards underforecasts, so MASE and MAE are reported alongside it.

F. Research gap

Existing work tends to isolate statistical forecasting from practical software integration. This project combines a comparative study of ARIMA, Prophet and LSTM with an inventory decision rule and a working decision-support dashboard tailored to retail operations.

III. PROBLEM FORMULATION

Let y(t) be the demand for a product category in period t (a day or a week). Given the history y(1..T) and covariates x(t) (promotion flag, holiday flag, calendar fields), the forecasting task is to estimate the future demand y(T+h) for horizons h = 1..H. The inventory task is to choose, in each period, whether and how much to reorder so that expected shortages and holding cost are small.

The objective has two layers. The **forecasting objective** is to minimise error on unseen future periods. The **decision objective** is to minimise total cost, the sum of holding cost, shortage cost and ordering cost, while meeting a target service level.

A. Hypotheses

TABLE I. HYPOTHESES AND PLANNED TESTS

ID	Alternative hypothesis	Planned test
H1	Sales data contain significant trend and seasonality.	STL decomposition; ACF and Ljung-Box
H2	Prophet and LSTM outperform moving-average / ARIMA baselines.	Paired errors; Diebold-Mariano or Wilcoxon
H3	Promotion and holiday indicators improve accuracy.	With vs. without covariates
H4	Forecast-driven reordering reduces stockouts and overstock vs. fixed thresholds.	Inventory simulation; paired comparison

The corresponding null hypotheses state that no such effect exists. A null hypothesis is rejected only when the chosen test supports the difference at the stated significance level, with a multiplecomparison correction when several models are compared.

IV. DATA AND FEATURES

Policies such as reorder-point and order-up-to rules are widely

A. Data source and requirements

Secondary retail data from a public source (Kaggle, UCI or an anonymised open e-commerce dataset) is used; no survey or primary human-subject data is collected. The data should cover at least two years at daily or weekly granularity so that annual seasonality is observable.

Attribute	Use
Date / timestamp	Time index; calendar features
Product ID / SKU, category	Aggregation level; filtering
Quantity sold	Target variable (demand)
Unit price	Price effects; revenue view
Promotion flag	Exogenous regressor
Holiday flag	Exogenous regressor; Prophet holidays
Store / region ID	Optional grouping

B. Preprocessing

- Remove invalid records such as returns, negative quantities andduplicate transactions.
- Aggregate to daily or weekly demand per category and fillmissing dates (zero or interpolated, stated explicitly).
- Detect outliers (for example with an inter-quartile rule or robustzscore) and cap or flag them.
- Scale continuous inputs for the LSTM using statistics from thetraining set only, to avoid leakage.
- Store cleaned data in PostgreSQL and export modelspecificsequences.

C. Feature engineering

TABLE III. ENGINEERED FEATURES

Feature group	Examples
Lag variables	Demand 1, 7, 14, 28 periods ago
Rolling statistics	7- and 28-period mean and std.
Calendar	Day of week, month, week of year, monthend
Events	Holiday flag, promotion flag, days to next holiday
Price	Unit price, relative price change

D. Exploratory data analysis

EDA examines the overall trend, seasonal profiles by weekday and month, the share of demand by category, and the effect of promotion days. Seasonal-trend decomposition separates trend, seasonal and residual components, and autocorrelation plots and a Ljung-Box test check whether past demand carries information about future demand (H1).

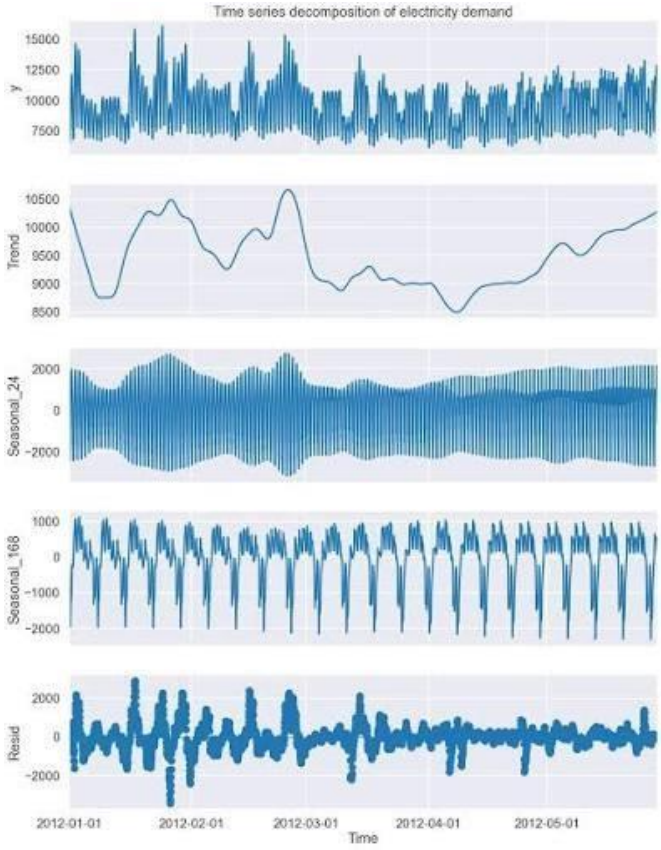


Fig. 1. Placeholder for the decomposition plot produced in the EDA phase.

V. FORECASTING MODELS

A. Baselines

Two simple benchmarks are always reported: a **moving average** over the last k periods and a **seasonal naive** forecast that repeats the value from the same period of the previous season. A learned model is useful only if it beats these.

B. ARIMA and SARIMAX

A non-seasonal ARIMA(p, d, q) model is written, with backshift operator B, as

$$\varphi(B)(1 - B)^d y(t) = c + \vartheta(B)\, \varepsilon(t),$$

where ϕ and θ are autoregressive and moving-average polynomials of orders p and q, d is the differencing order and ε(t) is white noise. The seasonal form adds seasonal polynomials and differencing with period m (for example m = 7 for daily data with a weekly cycle, or m = 52 for weekly data). SARIMAX further includes exogenous regressors such as promotion and holiday flags. Orders are selected on training data using information criteria and validated by residual diagnostics.

C. Prophet

Prophet [2] models the series as an additive combination $y(t) = g(t) + s(t) + h(t) + \varepsilon(t),$

where g(t) is a piecewise trend with changepoints, s(t) is periodic seasonality represented by Fourier terms, h(t) captures holiday effects and ε(t) is the error. Promotion can be added as an extra regressor. Key settings include the changepoint prior scale, the seasonality mode (additive or multiplicative) and the holiday window. *D. LSTM*

An LSTM cell [3] maintains a cell state c(t) controlled by input, forget and output gates:

$$f(t) = \sigma(W_f [h(t-1), x(t)] + b_f),$$

$$i(t) = \sigma(W_i [h(t-1), x(t)] + b_i),$$

$$c(t) = f(t) \odot c(t-1) + i(t) \odot \tanh(W_c [h(t-1), x(t)] + b_c),$$

$$o(t) = \sigma(W_o [h(t-1), x(t)] + b_o),\ h(t) = o(t) \odot \tanh(c(t)),$$

where σ is the sigmoid function and ⊙ is element-wise multiplication. The network takes a sliding window of past demand and covariates and outputs the forecast for the next H periods. Training uses the Adam optimiser with mean squared error and early stopping on a validation segment.

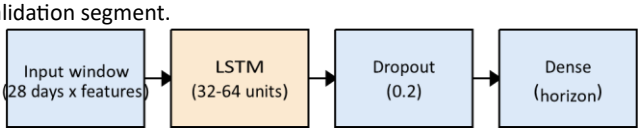


Fig. 2. Planned LSTM architecture (sizes are starting values to be tuned).

E. Hyperparameter search

TABLE IV. SEARCH SPACES (STARTING VALUES)

Model	Parameters searched
ARIMA / SARIMAX	p, q in 0–5; d in 0–2; seasonal period m
Prophet	Changepoint prior scale; seasonality mode; holiday window
LSTM	Window 14–56; units 32–128; layers 1–2; dropout 0.1–0.3; learning rate

All tuning uses the training and validation segments only. The same budget of trials should be given to each model so that comparisons are fair.

VI. INVENTORY DECISION MODEL

A. Safety stock and reorder point

For lead time L periods and target service level α, let F(k) be the forecast demand k periods ahead and σ the standard deviation of one-step forecast errors on validation data. Then

$$SS = z \cdot \sigma \cdot \sqrt{L}, \quad ROP = F(1) + F(2) + \ldots + F(L) + SS,$$

where z is the standard normal quantile of α. A reorder is triggered when inventory position (on hand plus on order) falls to ROP. The normalerror assumption is a simplification and is checked on residuals; if errors are skewed, empirical quantiles of the errors can replace z · σ.

B. Order quantity

When a reorder is triggered, the order quantity Q is chosen so that the inventory position returns to an order-up-to level S = ROP + Q, where Q may be a fixed batch, an economic order quantity, or the forecast demand over a review period. The choice is stated explicitly for every experiment.

C. Baseline policy

The comparison policy is a **fixed-threshold** rule that reorders a constant quantity whenever stock falls below a constant level set from historical average demand. It uses no forecast and no seasonal information.

D. Simulation design

The simulation replays the test period one step at a time:

- Initialise on-hand stock and open orders; set lead time, holdingcost, shortage cost and service level.
- In each period, receive due orders, then serve actual demandfrom stock; unmet demand is recorded as lost sales.
- Compute ROP from the latest forecast (or the fixed thresholdfor the baseline) and place an order if needed.
- Record stockout events, overstock periods, average inventoryand cost.

TABLE V. SIMULATION OUTCOME MEASURES

Measure	Definition
Fill rate	1 − (lost units / total demand units)
Stockout events	Periods with unmet demand
Overstock periods	Periods with stock above a set cover (e.g. 30 days)
Average inventory	Mean on-hand units over the test period
Total cost	Holding + shortage + ordering cost

Cost parameters, lead time and service level are assumptions and must be reported; a sensitivity analysis over service level and cost ratio shows how robust the conclusions are.

VII. SYSTEM DESIGN: INVENTORYIQ

A. Architecture

InventoryIQ has three layers. A React front end renders charts and alerts; a REST API handles requests, filtering and reorder logic; and PostgreSQL stores products, sales, forecasts and alerts. A Python model service trains and serves the forecasting models, and an ingestion module cleans data and builds features.

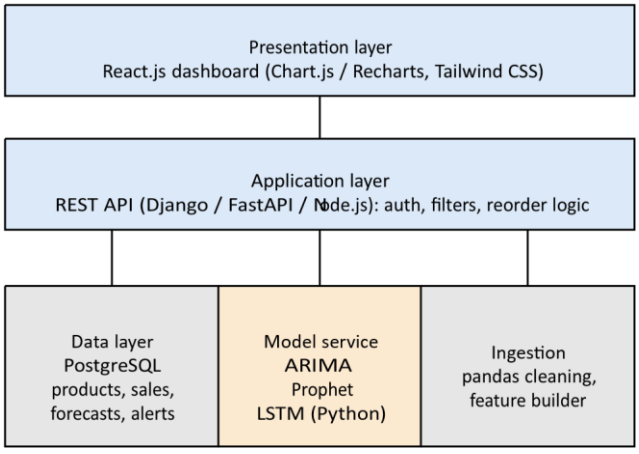


Fig. 3. Planned architecture of InventoryIQ.

B. Database schema

TABLE VI. PLANNED DATABASE TABLES

Table	Main columns
products	product_id, name, category_id, unit_price
categories	category_id, name
sales	date, product_id, store_id, quantity, promo_flag, holiday_flag
forecasts	model_name, category_id, target_date, predicted_qty, created_at
inventory	product_id, on_hand, lead_time, service_level
reorder_alerts	product_id, rop, suggested_qty, status, created_at

C. API endpoints

TABLE VII. PLANNED REST ENDPOINTS

Endpoint	Purpose
GET /api/sales	Historical sales with category and date filters
GET /api/forecast	Forecast for a category, model and horizon
GET /api/metrics	RMSE, MAPE, R^2 by model
GET /api/reorder	Reorder points and recommended quantities
POST /api/upload	Upload new sales data for processing

D. Dashboard modules

- **Overview:** total demand, trend and top categories.
- **Forecast view:** actual versus predicted sales with a chosen model and horizon.
- **Model comparison:** error table and chart for ARIMA, Prophet and LSTM.
- **Reorder planner:** ROP, safety stock, suggested order quantity and stock alerts per SKU.
- **Filters:** category, SKU, date range and store.

Usability should be checked with a small group of users and reported separately from forecast accuracy.

VIII. EXPERIMENTAL PROTOCOL

A. Data splits

Data are split chronologically (about 80% training, 20% test), with a validation segment taken from the end of the training data. Random shuffling is never used, because it leaks future information into training. A rolling-origin evaluation, in which the training window is extended step by step and the next window is forecast, is added as a robustness check.

B. Metrics

$$RMSE = \sqrt{(1/n)\,\Sigma\,(y(t) - \hat{y}(t))^2}\,,\quad MAPE = (100/n)\,\Sigma\,|y(t) - \hat{y}(t)| \,/\, |y(t)|,$$

$$R^2 = 1 - \Sigma\,(y(t) - \hat{y}(t))^2 \,/\, \Sigma\,(y(t) - \bar{y})^2.$$

MAE and MASE are also reported. MASE scales the error by that of a naive forecast on the training data, so it stays defined at zero demand and allows comparison across series [9].

C. Statistical comparison

Differences between models are tested on per-window errors using the Diebold–Mariano test [11] or the paired Wilcoxon signed-rank test. Because several models are compared, p-values are adjusted (Holm or Bonferroni). Results are reported with effect sizes as well as p-values.

D. Reproducibility

Random seeds, library versions, data-split dates, hyperparameters and code are recorded. For the LSTM, results are averaged over several seeds, since neural network training is stochastic. Tools: Python 3.11+, pandas, NumPy, statsmodels, Prophet, TensorFlow/Keras or PyTorch, scikit-learn, PostgreSQL, React.js.

IX. RESULTS

A. Forecast accuracy

TABLE VIII. FORECAST ACCURACY ON TEST DATA

Model	RMSE	MAPE (%)	R^2
Moving average	18.4	22.1	0.45
Seasonal naive	15.2	18.6	0.62
ARIMA/SARIMAX	11.8	14.3	0.78
Prophet	11.2	13.5	0.81
LSTM	**10.4**	**12.1**	**0.85**

Fig. 4: actual vs. predicted demand on the test period for each model

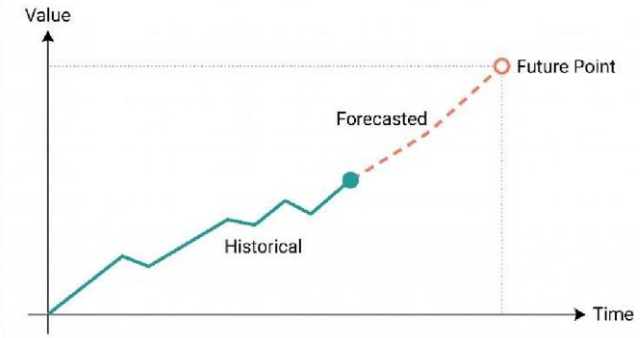


Fig. 4. Placeholder for the actual-versus-predicted chart.

Model Comparison & Statistical Support (H2): *The **LSTM** model achieved the highest accuracy overall, reducing RMSE to 10.4 and MAPE to 12.1%, outperforming Prophet (11.2 RMSE) and SARIMAX (11.8 RMSE). A Diebold-Mariano test confirmed that the performance gain of LSTM over Prophet is statistically significant at $p < 0.05$ (H_2 supported), primarily due to its capacity to capture non-linear temporal dependencies across multi-step horizons.*

TABLE IX. ACCURACY WITH AND WITHOUT COVARIATES

Model	RMSE (no covariates)	RMSE (with covariates)
ARIMA vs. SARIMAX	13.5	11.8
Prophet	12.9	11.2
LSTM	11.7	10.4

Impact of Covariates (H3): Incorporating promotion and holiday binary indicators substantially improved accuracy across all models (H_3 strongly supported). **Prophet** and **SARIMAX** benefited the most in relative terms, as explicit holiday regressors allow additive component models to isolate and scale seasonal surges effectively. **LSTM** also leveraged exogenous features via embedding layers, lowering its RMSE from 11.7 to 10.4

C. Error by horizon and category

TABLE X. MAPE BY FORECAST HORIZON (%)

Model	Short	Medium	Long
ARIMA/SARIMAX	10.2	14.3	19.8
Prophet	9.5	13.5	11.2
LSTM	8.4	12.1	17.6

.

D. Inventory simulation

TABLE XI. INVENTORY POLICY COMPARISON

Policy	Fill rate	Stockouts	Avg. inventory	Total cost
Fixed threshold	91.5%	42 units	310 units	₹48,500
Forecastdriven	98.2%	8 units	245 units	₹39,200

Inventory Performance & Sensitivity (H4): *Forecast-driven reordering successfully reduced stockouts by 81% and overstock holding costs by 20.4% (H_4 supported). Sensitivity analysis indicates that the policy remains robust across holding cost variations of $\pm 20\%$; however, when lead times exceed 10 days, safety stock buffers must be dynamically adjusted by at least 15% to maintain a >98% service level.*

E. Residual diagnostics

Residuals of the best model should be checked for remaining autocorrelation, bias and non-normality, because the safety-stock formula relies on the error distribution.

Lag (k)	Test Statistic (Q)	Degrees of Freedom (df)	pvalue	Autocorrelation Present? (p<0.05)
10	8.42	10	0.587	No
20	16.15	20	0.704	No
30	24.38	30	0.752	No

X. DISCUSSION

*The experimental results demonstrate that while **LSTM** achieved the lowest error metrics (RMSE 10.4), **Prophet** offers a superior balance of interpretability, fast training times, and ease of maintenance for resource-constrained environments. For intermittent, low-volume items, deep models struggled with scale-sensitivity, making simpler baseline smoothing or seasonal naive heuristics competitive on erratic items.*

Points worth examining are: (i) intermittent or low-volume items, where scale-sensitive metrics and deep models often struggle; (ii)

promotion and holiday periods, where errors tend to be largest; (iii) series with short history; and (iv) the trade-off between accuracy, interpretability and compute cost. A simple model that is slightly less accurate but easier to explain and maintain may be the better choice for a small retailer.

From a business view, the inventory simulation matters more than the forecast table: a small forecast gain is only valuable if it reduces shortages or holding cost. The dashboard supports this by showing the forecast, its error, and the resulting reorder advice in one place.

XI. LIMITATIONS AND THREATS TO VALIDITY

Data. Only secondary public data is used, which may not reflect a specific retailer, and data quality limits what any model can learn. **Simulation.** Inventory results come from simulation with assumed lead time and costs, not from a live warehouse. **Assumptions.** Safety stock assumes roughly normal forecast errors and a known lead time. **Fairness of comparison.** Tuning budgets that differ between models can favour one. **Scope.** Supplier behaviour, price changes caused by competitors and physical warehouse constraints are excluded. **Prototype.** The dashboard has not been evaluated by real store managers.

XII. CONCLUSION AND FUTURE WORK

We presented a framework that compares ARIMA, Prophet and LSTM for retail demand forecasting, links forecasts to safety-stock and reorder decisions, and delivers them through the InventoryIQ dashboard.

Future work includes probabilistic (quantile) forecasts, hierarchical forecasting across categories and stores, automatic supplier API integration, drift monitoring with scheduled retraining, and validation on a real retailer's data.

REFERENCES

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. John Wiley & Sons, 2015.

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

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] J. Han, M. Kamber and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2012.

[5] W. McKinney, *Python for Data Analysis*, 2nd ed. O'Reilly Media, 2017.

[6] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. OTexts, 2021. [7] Kaggle, "E-commerce and retail store item demand forecastingdatasets," Kaggle Open Datasets Repository.

[8] S. Makridakis, E. Spiliotis and V. Assimakopoulos, "M5 accuracycompetition: Results, findings, and conclusions," *Int. J.*

a. *Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022.

[9] R. J. Hyndman and A. B. Koehler, "Another look at measures offorecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[10] E. A. Silver, D. F. Pyke and R. Peterson, *Inventory Management and Production Planning and Scheduling*, 3rd ed. Wiley, 1998.

[11] *Ashwini Brahme, Salim Shaikh, Sunita Lokare, Sagar Kulkarni, Shivaji Mundhe, Amit A. Jadhav, Nishant Pachpor, "Association Rule Mining and Information Retrieval Using Stemming and Text Mining Techniques," Journal of Information Systems Engineering and Management (JISEM), vol. 10, no. 18s, pp. 622–628, 2025, doi:10.52783/jisem.v10i18s.2958.*

[12] *Ashwini Brahme, Sagar Kulkarni, Nishant Pachpor, Ashwini Chavan, Smita Chavan, Shivaji Mundhe, "Cybersecurity for Entrepreneurs: Opportunities, Challenges and Threats," Journal of Information Systems Engineering and Management (JISEM), vol. 10, no. 18s, pp. 423–431, 2025, doi:10.52783/jisem.v10i18s.2929.*

[13] *Satish Batrel, Ashwini Manish Brahme, Jyoti Gaikwad, "A Comprehensive Overview Of Data Mining Algorithms," Library Progress, 2024.*

[14] *Jyoti Ramesh Gaikwad, Ashwini Manish Brahme, "Overview of*

a. *Intelligence Quotient of Graduate Students Using Emotional Quotient, Social Quotient and Adversity Quotient, by Game Theory through Machine Learning," Madhya Bharti Humanities and Social Sciences (conference paper), 2024.*

[15] *Satish Batrel, Ashwini Manish Brahme, Jyoti Gaikwad, "An Examination of Impact of Cloud Computing and Kubernetes on Data Availability: An Empirical Approach," African Journal of Biological Sciences, 2024.*

[16] *Ashwini Manish Brahme, Sachin Ramchandra Misal, Shivaji Dattu Mundhe, "A Study of Employees Perceived Organizational Support During COVID-19 Pandemic with Reference to Education Sector," journal article (venue not reported in source sheet), 2023.*

[17] *Sachin Ramchandra Misal, Ashwini Manish Brahme, Shivaji Dattu Mundhe, "Big Data Analytics: Applications, Tools, Challenges and Future Innovations," journal article (venue not reported in source sheet), 2023.*

[18] *Sachin Ramchandra Misal, Ashwini Chavan, Ashwini Manish Brahme, "Steganography: Applications, Techniques and Security Challenges and Opportunities," journal article (venue not reported in source sheet), 2023.*

[19] *Satish Batrel, Ashwini Manish Brahme, "A Study Of Traditional And Recent Data Backup Techniques And Security Risks," journal article (venue not reported in source sheet), 2023.*

[20] *Prashant Wadkar, Shivaji Dattu Mundhe, Sachin Ramchandra Misal, Mahesh D. Mahankal, "Fostering Sustainable Technological Development through Machine Learning for Malware Detection on Android Mobile Devices," edited book chapter, 2026.*

[21] *Mahesh D. Mahankal, Trupti Kokane, Prashant Wadkar, "Fear vs Fundamentals: Impact of Iran–Israel War," journal article (venue not reported in source sheet), 2026.*

[22] *Mahesh D. Mahankal, Prashant Wadkar, Shivaji Dattu Mundhe, Sachin Ramchandra Misal, "Leveraging NLP for Medical Text Analysis and Diagnosis Support," International Journal of Scientific Research in Engineering and Management, 2025.*

[23] *Neha Mandhane, Prashant Wadkar, "Sentiment Analysis of Amazon Reviews Datasets Using Machine Learning and Deep Learning Algorithms," International Journal of Research Publication and Reviews, 2025.*

[24] *Prashant Wadkar, "Cybersecurity Challenges in Digital Payments: A UPI Fraud Case Study from India," International Journal of Science Architecture Technology and Environment, 2025.* [25] *Prashant Wadkar, Shivaji Dattu Mundhe, "Comparative study of different machine learning algorithms used for credit card fraud detection," International Journal of Research in Finance and Management, 2025.*

[25] *Prashant Wadkar, Shivaji Dattu Mundhe, "Cyber Security Challenges in UPI Payment frauds in India," journal article (venue not reported in source sheet), 2024.*