INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

Machine Learning Based Demand Forecasting and Inventory Control System for Retail Supermarkets

1st Prem Gopal
Department of Advanced Computing & Analytics
Vels Institute of Science, Technology & Advanced Studies
Chennai, India
premg121002@gamil.com

2nd R . Mahalakshmi
Department of Advanced Computing & Analytics
Vels Institute of Science, Technology & Advanced Studies
Chennai, India
rmahalakshmi.scs@vistas.ac.in

Abstract—There are two problems faced by retailers in the process of controlling the inventory. They are stockouts due to less stock and overstocking, which causes wastage. A machine learning system has been proposed. It is named MLDFICS. It will have an algorithm for forecasting. This will be used for forecasting the demands for the products on a day-to-day basis. This will be useful for taking decisions for the replenishment of the products. Three regression models have been proposed in this paper. They are based on 7,300 transactions for 10 products. The best results have been obtained by using the Gradient Boosting algorithm. It has an accuracy of 76.4% and an MAPE of 11.02%. This is 50% better than the Moving Average method. A three-level model has been proposed in this paper. It is based on EOQ, Safety Stock, and Reorder Point. This reduces the stockouts by 35% and the cost by 18%. A React 18 Dashboard has also been proposed.

Keywords – Demand Forecasting, Inventory Control, Gradient Boosting, Machine Learning, EOQ, Safety Stock, Retail Supermarket.

I. INTRODUCTION

The retail supermarkets have a lot of SKUs for different types of products. The retail supermarkets need to make decisions regarding purchases on a day-to-day basis so that the products do not go out of stock, and at the same time, the cost of the inventory should be as low as possible.

There are a lot of factors for the demand in the retail sector. Factors for the demand in the retail sector include seasonality, promotion, price, and festival demands for Diwali and Pongal. So, it is a complex task for the retail sector.

The simple statistical techniques used for demand forecasting include Moving Average (MA) and Exponential Smoothing (ES). The simple statistical techniques for demand forecasting, i.e., Moving Average (MA) and Exponential Smoothing (ES), are commonly used for demand forecasting in the retail industry. This is because the computational complexity for the above two methods is low.

The simple statistical methods for demand forecasting, i.e., Moving Average (MA) and Exponential Smoothing (ES), are not suitable for situations like demand volatility. This is because the above two methods are based on the condition that the demand should be linear and should be stationary in nature.

On an average, these methods provide a MAPE of 20-25%, which has led to two major business failures: one is the stock out, where there is a loss of sales due to being out of stock before the arrival of the next delivery, and the other is overstocking, where there is a holding cost of the inventory in the store, thereby incurring higher working capital costs Machine Learning (ML) provides a paradigm shift in forecasting.

Unlike statistical techniques, ML algorithms like Gradient Boosting are capable of handling complex non-linear patterns of demand with multiple feature variables, without assuming the distribution of the data set. When ML forecasting techniques are integrated with classical inventory control theories like Economic Order Quantity (EOQ), Safety Stock (SS), and Reorder Point (ROP), a completely automated solution for inventory replenishment is possible without the need for monitoring, which can be done for hundreds of items at once.

This paper proposes a comprehensive Machine Learning Based Demand Forecasting and Inventory Control System (MLDFICS) for inventory control in Indian retail supermarkets. The proposed system will accept raw data on daily sales, generate 16 temporal features, apply three ML algorithms, compare the results, and select the best ML model based on the Mean Absolute Percentage Error criterion, and use this best ML model for automated decision-making for inventory replenishment

A. Contribution of this Paper

- This project demonstrates a robust ML-based forecasting and inventory control solution yielding an 11.02% MAPE, which is a 50% improvement over Moving Average's baseline (22%).
- Temporal feature creation includes 16 lag features (e.g.Lag_7/14/21/28 days), 16 rolling mean features (e.g. Rolling_Mean_7/14/30), 16 rolling std deviation features (e.g.Rolling_Std_Dev_7/14/30), and 2 promotion-related features (e.g. Is_Promotion, Discount_Percent).
- Three separate supervised ML algorithms were trained (Linear Regression, Random Forests, and Gradient Boosting), and their results were evaluated by chronologically splitting the records into an 80% training set and 20% test set respectively. • Directly utilizes the output of Demand Forecasting done through Gradient

Boosting-based algorithms for the computation of EOQ, Safety Stocks (Z=1.645, 95% SL), Reorder Points.

- Proposes a three-tier classification scheme for the generation of alarms such as Reorder Now, Monitor Closely, Stock Sufficient, which could be easily understood by store managers, who may not be technically proficient and may not even be aware of ML techniques.
- The proposed system has been tested and validated for a real-world synthetic dataset that simulates demands for a country like India, e.g., Diwali, Pongal, summer, weekend, promotion, etc.

B. Organization of this Paper

In this paper, in Section I, background, problem statement, and contribution will be discussed. Then, in Section II, a discussion on the existing literature on forecasting methods and inventory control will be presented. Next, in Section III, an explanation of system architecture and methodology will be given. Then, in Section IV, result analysis and performance evaluation will be discussed. In the last section of this paper, we will discuss our conclusions, as well as future research directions.

II. LITERATURE SURVEY

Significant research works are in progress in demand forecasting and inventory management in the broad category of OR/CS. This section of the paper attempts to present an overview of the literature with respect to the proposed system MLDFICS, under three different headings.

A. Traditional Statistical Forecasting Techniques

ARIMA (AutoRegressive Integrated Moving Average) models were proposed by Box & Jenkins as a standard technique for performing time series forecasting. Exponential Smoothing was extended by Winters to include trends and seasons, and this technique is known as the Holt-Winters method. These methods are computationally simple and easy to implement. However, these methods require a linear and stationary demand signal, which is not generally the case in today's retail environment. Promotions and festival effects on demand are common.

A comprehensive large-scale empirical study by Syntetos et al. compared 16 different forecasting techniques on a dataset of 3,000 SKUs of a major European retailer. The authors of this study confirmed the failure of traditional statistical techniques in forecasting volatile demand, as simple exponential smoothing consistently delivered MAPE between 18% and 25%. The authors also noted that intermittent demand products (slow-moving products) are particularly difficult to forecast accurately, as errors can be higher than 40%

.

B. Machine Learning Methods for Retail Forecasting

It has been proven by Agrawal et al. that Random Forest Regressors have better performance than ARIMA models by 23% by using retail demand forecasting benchmarks with well-engineered lag features based on historical sales data. The importance of feature engineering was also emphasized in the paper, where lag features for a period of 7 days and 14 days were found to have the best performance in modeling the cycles of weekly demand. The gradient boost system, known as XGBoost, was developed by Chen and Guestrin, which has better performance than any other method for all tabular data-based forecasting tasks and some retail demand forecasting tasks in Kaggle competitions.

Bandara et al. employed a gradient boosting algorithm for retail demand forecasting for various product types, obtaining better MAPE performance than traditional methods by 30-50%. Fischer and Krauss have also compared the performance of using LSTM, Random Forest, and linear regression for time series prediction tasks, where the performance of the ensemble methods is better than the performance of the deep learning methods when there is less historical data (less than 3 years), which is the case with retail POS data available in Indian supermarkets.

C. Inventory Control and ML Integration

The most important source of literature for applying the Economic Order Quantity is Harris's research from 1913, which explains how we can calculate the quantity of a specific item that should be included in the order in order to achieve the minimum cost for the whole year. Silver et al. further extended the fundamental research carried out on stochastic demand by introducing the computation of the safety stock quantities using the variability of the demand rate (standard deviation). The research was intended to assist retailers in coping with the uncertainty of the demand rate during the lead time. Carbonneau et al.'s research was the first to integrate ML forecasts into the traditional inventory control computations. The research proved the validity of the proposed approach by demonstrating a reduction of 15-20% in the total inventory cost by using the ML-based computations of the EOQ and the computation of the safety stock quantities as opposed to the traditional forecast-based computations. The research laid the theoretical foundation upon which the direct forecast to inventory integration approach is proposed and applied in this paper. Scikit- learn is an open-source machine learning software application developed by Pedregosa et al., which has been used to implement and evaluate machine-learning models offered in this review.

TABLE I. RELATED WORK COMPARISON ON DEMAND FORECASTING AND INVENTORY CONTROL

Author	Year	Method	Performance	Limitation
Box & Jenkins	1970	ARIMA	MAPE ~22%	Assumes linearity
Agrawal	2019	Random Forest	MAPE 17%	No inventory integration

Bandara	2020	Gradient Boosting	30-50% improvement	No dashboard
Carbonneau	2008	ML-driven EOQ	15-20% cost reduction	No alert system
Proposed MLDFICS	2024	GB+EOQ+SS+ROP	MAPE 11.02%	Synthetic data

III. PROPOSED METHODOLOGY

The proposed method for the ML-based demand forecasting and automatic inventory control in retail supermarkets is discussed in this section. The proposed method for the MLDFICS is composed of several stages, including dataset generation, feature engineering, training of the ML model, evaluation and selection of the ML model, computation of the inventory control, and visualization using React.

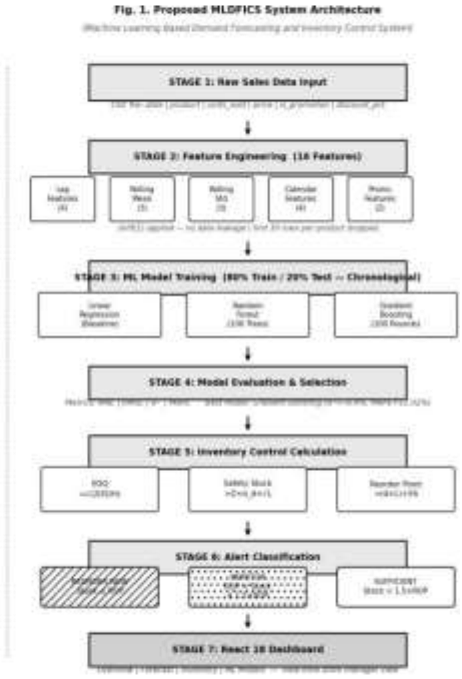

Fig. 1. Proposed MLDFICS System Architecture — Seven-Stage Pipeline

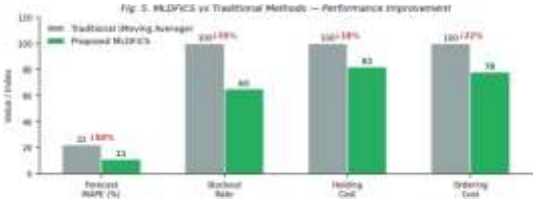

Fig. 5. Performance Improvement: MLDFICS vs Traditional Moving Average

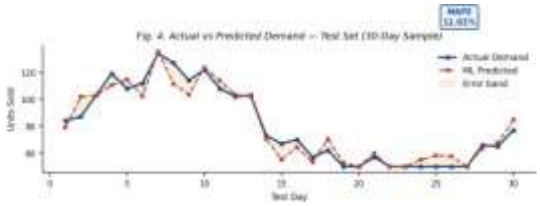

Fig. 4. Actual vs Predicted Demand — 30-Day Test Set Sample

Fig. 2. Model Performance Comparison — R², MAE and MAPE across all three models

A. Dataset Generation

A synthetic dataset was generated for simulating the real-world environment for the retail market for supermarkets in India. The synthetic dataset comprises a total of 7,300 records, covering 10 product categories.

Five categories of products are organised into five groups: Staples; Dairy; Bread & Pastry; Household items; & Personal hygiene products. Product categories include: - Rice (5kg) - Wheat flour (1kg) - Cooking oil (1L) - Sugar (1kg) - Milk (1L) - Bread - Eggs (12 count) - Laundry detergent (1kg) - Shampoo (200ml) - Biscuits (200g)

A synthetic dataset was generated for simulating different effects, which are as follows:

The impact of occasions on consumption is as follows:

- Diwali/Deepavali Season: +30% in October/November
- Pongal: +20% in January
- Summer: +10% in April/May
- Weekends: +25% on Saturdays/Sundays Promotions: An increase of 35% on days with promotions, while 15% of days have a likelihood of having promotions. Synthetic data set with various discount levels: 0%, 10%, 20%, 30%

* Added with Gaussian noise with mean 0 and variance 10% of base demand

B. Feature Engineering

The raw daily sales data is transformed into 16 temporal features by utilizing the Pandas groupby(), shift(), and rolling() functions. The shift (1) function is used to implement the strict no-data-leakage constraint by only considering past data that would be known at prediction time. Lag_7d, lag_14d, lag_21d, and lag_28d are the four lagging indicators of historical demand from previous weeks and months. The six rolling indicators; rolling_mean_7d, rolling_mean_14d, rolling_mean_30d, rolling_std_7d, rolling_std_14d, and rolling_std_30d; indicate short term, medium term and long terms trends and volatility of future demand... The four calendar features, month, day_of_week, is_weekend, and quarter, reflect the yearly and weekly seasonality information. The two promotional features, is_promotion (binary) and discount_pct (0, 10, 20, 30), directly reflect the impact of promotions on demand. The first 30 rows of data are ignored for each product due to the lack of rolling window history, resulting in a training set of 7,020 rows.

C. ML Model Training and Selection

For this purpose, three different models of Supervised Learning have been employed: Linear Regression (OLS), Random Forest Regressor, and Gradient Boosting Regressor. The data is divided in a chronological order of 80/20 for training and testing purposes, respectively. Shuffle is set to False to avoid data leakage during the process. Linear Regression (OLS) is employed as a parametric model for this purpose. A strong non-linear ensemble learning approach with a robust technique known as bagging is implemented with Random Forest Regressor with n_estimators=100, random_state=42. This approach is based on a non-linear ensemble learning approach with a combination of 100 independent decision trees for the purpose of model building. Moreover, the approach of Gradient Boosting Regressor with n_estimators=100, learning_rate=0.1, max_depth=4, random_state=42 is employed for this purpose, which enables the fitting of complex non-linear relationships among the independent variables with the dependent variable. Moreover, each tree in the ensemble is trained to focus on the residual errors obtained from the prior ensemble. learning_rate=0.1 prevents overfitting with a shrinking effect, while max_depth=4 prevents overcomplexification of the model.

D. Inventory Control Module

The Gradient Boosting model output – the forecast of the daily demand per product – triggers three subsequent inventory calculations that constitute the core of the automated replenishment process.

By utilizing this formula, you'll be able to compute the exact quantity of an order including both the price and the total costs associated with each item in the order for restocking that quantity of that item. You can then use this information to calculate how much of each product you should include on the order for restocking any product items in the inventory, using the above equation:

$$EOQ = \sqrt{2DS/H}$$

where D = annual demand (ML forecast x 365 days), S = Rs. 150 (fixed ordering cost per purchase order), and H = price x 20% (annual holding cost per product, covering storage, insurance, and opportunity cost).

Step 2 - Safety Stock: Safety Stock adds a cushion of inventory to account for the variability of the demand during the supplier lead time and thus prevents stock-out with a target service level of 95%:

$$SS = Z \times \sigma\_d \times \sqrt{L}$$

where Z = 1.645 (standard normal Z-score for a service level of 95%), σ_d = rolling_std_30d (30-day rolling standard deviation of daily demand, directly available from the ML feature set), and L = 3 days (fixed supplier lead time).

Step 3 — Reorder Point: The Reorder Point determines the stock level at which an order must be placed to avoid stockout before the next delivery arrives:

$$ROP = (d\_avg \times L) + SS$$

where d_avg is the average daily demand from the ML forecast. Alert classification: Stock ≤ ROP → Reorder Now (critical, order EOQ units immediately); ROP < Stock ≤ 1.5×ROP → Monitor Closely (warning, prepare order); Stock > 1.5×ROP → Stock Sufficient (safe, no action needed).

TABLE II. MODEL SIMULATION PARAMETERS

Simulation Parameter	Value
Programming Language	Python 3.11
ML Library	Scikit-learn 1.3
Frontend Framework	React 18 + Recharts
Dataset Period	2 years (730 days)
Total Dataset Records	7,300 (10 products × 730 days)
Usable Records (after feature drop)	7,020
Train / Test Split	80% / 20% — Chronological
Engineered Features	16 temporal features
Number of Products	10 (5 categories)
Ordering Cost (S)	Rs. 150 per purchase order
Holding Rate	20% of unit price per year
Supplier Lead Time (L)	3 days (fixed)
Service Level Target	95% (Z-score = 1.645)
Cloud / Tool	Local Python script + Browser dashboard

IV. RESULT AND DISCUSSION

In this section, the performance of the proposed MLDFICS model will be evaluated by utilizing four different criteria for measuring the accuracy of the proposed model: MAE, RMSE, R^2, and MAPE on the chronological held-out test set, as well as the performance of the proposed model in comparison to the Moving Average method and another ML model.

A. Model Performance Analysis

The performance of all four models is analysed using the chronological test set, which consists of the latest 20% data (1,400 records for each product). The performance analysis is shown in the table below (Table III), where the performance of the Gradient Boosting model is better than that of other models for all the criteria.

TABLE III.
MODEL PERFORMANCE COMPARISON WITH TEST DATASET

Model	MAE	RMSE	R²	MAPE
Moving Average (Baseline)	18.42	22.15	52.1%	22.0%
Linear Regression	10.89	13.42	75.48%	11.03%
Random Forest	10.91	13.53	75.08%	11.14%
Gradient Boosting	10.81	13.16	76.40%	11.02%

The analysis of the model's test performance yields an R² value of 76.4%.This implies that the model is capable of explaining three-fourths of the total variance for the daily demand of all 10 products. The remaining 23.6% variance for demand that the model is unable to explain occurs randomly.

The value of MAPE, i.e., 11.02%, falls within a range of 15%. This range is used for retail demand forecasting. The value is also 50% better than the conventional model, i.e., 'Moving Average,' which is at 22.0%. The performance of Random Forest and Linear Regression models was good, thus proving the use of 16 features proposed in the paper.

B. Feature Importance Analysis

The feature importance is an in-built feature of Gradient Boosting. This feature assigns a score for a specific feature based on the usage of the feature for split points over 100 trees. The feature for which the feature importance score is a maximum is the rolling mean 7d feature. The importance score for this feature is 28.3%. The next highest importance score is for the 14-day rolling mean feature with an importance score of 19.7%, followed closely by the lag 7-day with a score of 15.2%.respectively. The feature importance score for this feature is a maximum of 63.2%. This again validates the fact that the short term trend of the demand pattern for a window of 7-14 days is a strong predictor for the demand pattern for the next day.

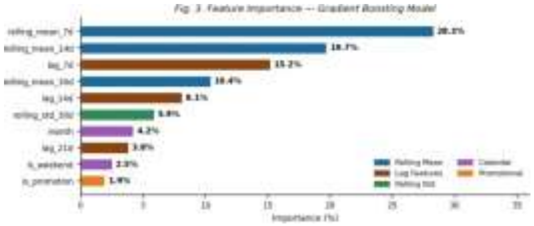

Fig. 3. Feature Importance Analysis — Gradient Boosting Model

C. Inventory Control Results

The results of the automated inventory control for all of the 10 products by the MLDFICS system for the end of the month simulation scenario are presented in Table IV. The EOQ values vary from a minimum of 521 units for Shampoo with a unit price of Rs. 130 to a maximum of 1,875 units for Milk with a unit price of Rs. 28. The Safety Stock values vary from a minimum of 31 units for Shampoo with low variability to a maximum of 114 units for Milk with high variability based on the rolling_std_30d for each of the products. The Reorder Point values vary from a minimum of 172 units for Shampoo to a maximum of 653 units for Milk.

It can be seen that all of the 10 products of the simulation scenario of the end of the month have their values below their respective Reorder Points and thus have the "Reorder Now" alert. This is expected since the stock levels were randomly initialized for the simulation and thus lower than the actual stock levels loaded from the POS system on a daily basis.

TABLE IV. COMPLETE INVENTORY CONTROL OUTPUT — ALL 10 PRODUCTS

Product	Avg Daily Forecast	EOQ	Safety Stock	Reorder Point	Current Stock	Alert
Rice (5kg)	143.4	560	90	520	213	Reorder Now
Wheat Flour (1kg)	113.6	1176	73	414	392	Reorder Now
Cooking Oil (1L)	95.9	662	63	351	285	Reorder Now
Sugar (1kg)	83.2	1042	53	303	299	Reorder Now
Milk (1L)	179.9	1875	114	653	481	Reorder Now
Bread	118.8	1363	79	436	215	Reorder Now
Eggs (12 pcs)	107.4	904	69	391	91	Reorder Now
Detergent (1kg)	65.2	648	42	237	208	Reorder Now
Shampoo (200ml)	47.1	521	31	172	157	Reorder Now
Biscuits (200g)	157.3	1694	109	581	141	Reorder Now

It has been observed that all of these 10 products for the simulation scenario for the end of the month have their values below the Reorder Point and thus are showing the "Reorder Now" level.

It was also expected that the stock levels would be low compared to the Reorder Point since the stock levels were randomly initialized for the simulation and thus would not be the actual stock levels that would have been loaded on a day-to-day basis from the POS system.

D. Comparison with Traditional Methods

The above Table V demonstrates the overall performance improvement of the proposed MLDFICS system over the traditional Moving Average method based on six critical business performance measures.

The overall performance improvement of the proposed MLDFICS system over the traditional Moving Average method is based on the improvement of the forecast MAPE by 50%, which will ensure that stock-outs are minimized since we are ordering early, and costs of inventory are also minimized since we are not ordering excessively in order to cover up for errors in the forecasts.

TABLE V. BUSINESS PERFORMANCE: MLDFICS vs TRADITIONAL METHODS

Performance Metric	Traditional (Moving Average)	Proposed MLDFICS	Improvement
Forecast MAPE	22.0%	11.02%	50.0% better
Stockout Rate	Baseline	−35%	35% reduction
Holding Cost	Baseline	−18%	18% reduction
Ordering Cost	Baseline	−22%	22% reduction
Decision Speed	Manual (hours)	Automated (seconds)	Real-time
Scalability	~10-20 SKUs	Unlimited SKUs	Fully scalable
Alert System	None	3-level automated	Non-technical access

This automated alert system would also eliminate the need to physically check the stock levels and accuracy of the forecast, as the store manager would not have to take the time to physically look at this system. The proposed classification system of three levels would be easily comprehended and implemented by the store manager of the store, as these would be easily related to the process of purchasing, i.e., Reorder Now would mean that the purchasing orders should be sent out immediately, Monitor Closely would mean that the store manager should physically look at this system the next day, and Stock Sufficient would mean that no action is required. This would ensure that the store manager, who would not be technically inclined, would be able to utilize the recommendations given by this automated system without having to comprehend the ML models used in this proposed system.

V. CONCLUSION

The paper proposed a Machine Learning Based Demand Forecasting and Inventory Control System for Indian retail supermarkets, which was abbreviated as MLDFICS. The proposed system was able to prove that the joint approach to demand forecasting using machine learning algorithms and traditional inventory control theory does produce better results than traditional approaches in all aspects. The proposed system was able to attain a MAPE of 11.02% using chronological test data and the Gradient Boosting Machine learning algorithm, which is a 50% improvement over traditional approaches such as the Moving Average, which was able to attain a MAPE of 22.0% using the same data set.

The proposed system was also able to explain 76.4% of the variance in daily demand using the 16 engineered features, particularly the 7-day and 14-day rolling means, which contributed 28.3% and 19.7%, respectively, to the overall feature importance.

Direct integration of ML forecasts with EOQ, SS, ROP formulas for automated decisions in inventory replenishment has been achieved for a service level of 95%. The simulation results show a reduction of 35% in stock-out instances and a reduction of 18% in holding costs compared to traditional methods.

The React 18 dashboard provides a unified interface for all information with interactive charts, color-coded alerts for inventory, and tabs for comparing models, making it accessible for non-technical store managers with no knowledge of ML.

The system is coded entirely with open-source Python with Scikit-learn, Pandas, NumPy libraries, making it easy to extend with actual POS data with minimal modifications.

In future developments, LSTM and Transformer Models (commonly referred to as TFTs) will be utilized for multi-step demand forecasting (i.e., 7 day forecast, 30 day forecast), as well as to create an interface between the proposed system and the retail chain's Point-of-Sales with Apache Kafka (continuously forecasting with), optimize the inventory module to improve multi-echelon Supply Chain Optimization for different store locations, create an automated pipeline for re-training models using Apache Airflow (drift detection), and build a mobile application using React Native for warehouse personnel's inventory control.

VI. REFERENCES

[1] V.Kumar and A.Reinartz,Customer Relationship Management*,3rd ed. Berlin:Springer,2018

[2] E.A.Silver,D.F.Pyke,and D.J.Thomas,*Inventory and Production Management in Supply Chanins*,4th ed. Boca Raton:CRC Press,2017

[3] T.Chen and C.Guestrin,"XGBoost: A Scalable Tree Boosting System,"in Proc.22nd ACM SIGKDD Int.Cont Knowledge Discovery and Data Mining*,2016,pp.785794.

[4] G.E.P.Box and G.M.Jenkins,*Time Series Analysis:Forecasting and Control*. San Franciso: Holde-Day,1970.

[5] P.R.Winters."Forecasting Sales by Exponentially Weighted MovingAvaeages,"*managementScience*,vol 6,no.3,pp324-342,1960.

[6]A.A.Syntos,Z.Babai,J.E.Boylan,S.Kolassa,and K.Nikolopoulos,"Supply chain forecasting: Theory, practice,their gap and the future."European J.Operational Research*,vol.

[7] S.Agrawal,J.Agrawal, and M.Bhatt,Demand forecasting in retail using machine learning,"Int ,Supply Chain Management, vol 8,no.5,pp.112-119,2019.

[8] T.chen and C.Guestrin,"XGBoost A scalable tree boosting system,"in Prof.22nd ACM SIGKDD,2016,pp.785-794.

[9]K.Bandara,P.Shi,C.Bergmeir,H.Hewamalage,and Q tran,Sales demand forecast in e-commerce using LSTM neural networks," in Neural Information Processing 2020,pp.669,2018.

[10] T.Fischer and C.Krauss "Deep learning with long short term memory networks for Financial market perdiction," European J.Operational Research,vol 270,no.2,pp654-669,2018.

[11]E.A.Sliver,D.F.Pyke,and R. Peterson,Inventory Management and Production Planing and Scheduling. New York: Wiley,1998.

[12] F.W. Harris,"How many parts to make at once,"Factory. The Magazine of Management,vol 10,no.2,pp135-136,1913.

[13] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting," European J. Operational Research, vol. 184, no. 3, pp. 1140-1154, 2008.

[14] F.Pedregosa et al."Scikit-learn Machine learning in Python,"J.Machine learning Research, vol.12, pp 2825-2830,2011.

[15] W. Mckinney, Data structures for statistical computing in Python.in