


PAPER

XGBoost-Based Demand Forecasting in Supply Chain Management Using Machine Learning Algorithm

S. Nagadevi¹(✉),
G. Abirami¹, R. Vidhya¹,
S. Selvakumar² ,
C. Vijayalakshmi³

¹Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India

²Department of Computer Science and Business Systems, Panimalar Engineering College (Autonomous), Chennai, Tamil Nadu, India

³Department of Computer Science and Engineering, B S Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India

nagadevs@srmist.edu.in

ABSTRACT

Effective supply chain management (SCM) plays a significant role in enterprises seeking to mitigate risks, optimize overall productivity, and decrease costs while maintaining product quality and customer satisfaction. Addressing quality defects within the supply chain proactively assists in minimalizing returns, recalls, and rework, which results in increased profitability and considerable financial benefit. For the accurate delivery of products and collecting feedback for analysis, enterprises rely on SCM to satisfy customer needs. Machine learning (ML) approaches have witnessed a revolutionary milestone in SCM, facilitating more effective management of the supply chain. However, operational costs, model interpretability, and data quality remain to be a major challenges. Therefore, this paper presents XGBoost-based demand forecasting in SCM using an ML algorithm. At the initial stage, Z-score normalization servers as a data preprocessing to normalize the input data features into uniform scales. For the feature selection process, the genetic algorithm (GA) is exploited to identify the most descriptive variables. Then, the XGBoost model is employed for the demand forecasting process to accurately forecast future market trends. Finally, particle swarm optimization (PSO) involves hyperparameter tuning which allows the model to achieve optimal forecasting results. Experimental outcomes highlight that the proposed technique accomplishes superior forecasting performance compared to other existing approaches.

KEYWORDS

supply chain management (SCM), demand forecasting, machine learning (ML), XGBoost, particle swarm optimization (PSO)

1 INTRODUCTION

In today's dynamic and competitive business environment, effective supply chain management (SCM) is a critical determinant of operational success [1]. SCM is the process of coordinating and integrating the workflows of goods, information, and finances throughout all businesses involved in product or service delivery to the customer.

Nagadevi, S., Abirami, G., Vidhya, R., Selvakumar, S., Vijayalakshmi, C. (2025). XGBoost-Based Demand Forecasting in Supply Chain Management Using Machine Learning Algorithm. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(18), pp. 161–174. <https://doi.org/10.3991/ijim.v19i18.57253>

Article submitted 2025-04-28. Revision uploaded 2025-06-19. Final acceptance 2025-06-21.

© 2025 by the authors of this article. Published under CC-BY.

Enterprises largely depend on the efficiency of their SCM. Contemporary supply chains are unpredictable and complicated; besides conventional approaches face challenges in adapting to these factors. Increasing the efficiency of the supply chain through better-quality demand prediction, supplier performance, streamlined logistics, and inventory optimization processes could be accomplished with machine learning (ML) and sophisticated data analytics methods [2–15]. For making informed decisions about delivery, production, and distribution processes, along with controlling the flow of products, SCM can be strengthened by mathematical modelling. Mathematical model provides companies with the ability to evaluate data, minimize costs, increase productivity, and ultimately upsurge profitability. The efficient movement of goods in a supply chain relies on the ability to accurately forecast product demands [3]. These forecasts often produced within a hierarchical structure which might characterize time periods or geographic regions, product families. The organizations are striving to flawlessly incorporate the operations and processes of their supply chain to enhance the overall productivity. This transformation can be observed across industries, including healthcare, manufacturing, retail, etc. [4].

Determining safety stocks and demand forecasting are major components of supply chain planning. This includes predicting future demand for the products or services using past statistics and other external and internal drivers. Excess production and stockouts may be decreased by precisely forecasting demand. This enables businesses to better planning of logistics, production, and inventory. Firms ensure safety stocks in their inventory to defend against unexpected fluctuations in supply or demand. In order to satisfy the customer needs while balancing decreased carrying and inventory costs, organizations should find the suitable safety stock level. This helps to minimize production costs, optimize inventory, and improve customer service [5].

By spotting trends and analyzing past data, ML methods play a critical role in demand forecasting. Interest in resilience and adaptability in the face of shifting market dynamics has increased as a result of the expanding use of ML in SCM. To ensure optimal inventory placement across its extensive transportation network, Amazon, for example, employs ML algorithms to estimate consumer orders based on seasonality, purchasing behavior, and browsing history. By placing products closer to anticipated demand areas, Amazon is able to significantly reduce transportation costs and delivery times thanks to these ML-powered forecasting abilities. Likewise, ML-based forecasts help stores dynamically adjust inventory levels, aligning supply with fluctuating customers' requirements in the retail sector [6].

In this paper, we introduce an XGBoost-based demand forecasting in SCM using an ML algorithm. At the initial stage, Z-score normalization serves as data preprocessing to normalize the input data features into uniform scales. For feature selection process, the GA model is exploited to identify the most descriptive variables. Then, the XGBoost model is employed for the demand forecasting process to accurately forecast future market trends. Finally, PSO algorithm involves hyperparameter tuning which allows the model to achieve optimum forecasting outcomes. Experimental outcomes highlight that the proposed technique achieves higher efficiency than other existing methodologies.

2 LITERATURE REVIEW

Aliahmadi [7] introduces an ML-based architecture for improving multi-objective supply chain networks, which primarily focuses on optimizing decision-making in

dynamic and complex environments. The presented technique incorporates risk assessment, predictive analytics, and clustering by using cutting-edge ML algorithms to improvise numerous supply chain operations, such as transportation planning, demand forecasting, and inventory management.

Verma [8] inspects how AI is transmuting SCM by means of dynamic optimization, demand forecasting, and real-time inventory management, through insights from McKinsey's 2023 report. AI-based solutions such as digital twins and ML have greatly enhanced service delivery, logistics costs, and inventory levels. The study emphasizes the strategic significance of aligning AI implementation with business goals to effectively resolve the difficulties of supply chain.

Nweje and Taiwo [9] incorporate AI with IoT and block chain (BC). IoT device tracks real time items, enabling proactive disruption management and improving transparency. However, BC fosters stakeholder trust and collaboration and maintains data integrity across the supply chain. The study displays that the leading companies have succeeded in integrating AI into their supply chains, improving demand forecasting accuracy and operational cost reductions.

Zain [10] examines the effect of ML-based supply chain digital twins on three significant aspects: operational performance, responsiveness, and sustainability. In this work, ML approaches allow prescriptive decision-making and predictive analytics within the digital twins to align supply chains with sustainability goals, minimize waste, and enhance resource usage. Moreover, it investigates how digital twins optimize supply chain agility by simplifying processes, fostering resilience, and progressing its quick response to disruptions. Finally, the operational benefits of ML-empowered digital twins, including better collaboration between stakeholders, higher forecasting accuracy, and enhanced resource allocation, can be investigated.

Lu et al. [11] investigate the applications of intelligent technology in supply chain demand forecasting and inventory management, emphasizing how ML, big data, and AI technologies can increase inventory management efficiency and forecasting accuracy. With the comprehensive analysis, this study proposes corresponding inventory management optimization strategy and illustrates the implementation effect of intelligent demand forecasting techniques.

Mahin et al. [12] offer a state-of-the-art technique to sales prediction through ML approaches to optimize the predictive accuracy and improving supply chain operations after analyzing past sales data and considering various factors such as stock, seasonality, and trends. Numerous ML methods were employed, namely the ensemble voting regressor, linear regression, elastic net regression, KNN, and random forest (RF). The RF and KNN shows lesser accuracy but the voting regressor achieved higher performance than other techniques.

3 PROPOSED METHODOLOGY

In this work, we present XGBoost-based demand forecasting in SCM using an ML algorithm. At the initial stage, Z-score normalization serves as a data preprocessing to normalize the input data features into uniform scales for maximizing model stability. For feature selection process, the genetic algorithm (GA) is exploited to identify the most descriptive variables. Then, the XGBoost model is employed for the demand forecasting process to accurately forecast future market trends. Finally, particle swarm optimization (PSO) involves hyperparameter tuning which allows the model to achieve optimum forecasting outcomes [13].

3.1 Z-score normalization

In the initial phase, data preprocessing can be done using the Z-score normalization to normalize the input data into uniform scales. In SCM [14], the supplier data have varied scales. If the clustering is carried out using raw data, then the accuracy of the data might be decreased. To resolve this problem, Z-score normalization is used to convert all data into uniform scale and it can be mathematically formulated as follows:

$$Z = X - \frac{\mu}{\sigma} \tag{1}$$

In Eq. (1) the normalized and supplier logistic values of corresponding data values are Z and X. The mean of the corresponding data in the supplier logistics is μ . The working process of the propose method is shown in Figure 1. The standard deviation of the supplier logistics value is σ . The mean is 0 and standard deviation is 1 and it can be represented by the following equation:

$$\sigma = \sqrt{(\Sigma(X - \bar{X}) / n)} \tag{2}$$

In Eq. (2), the supplier logistics value of corresponding feature is denoted by X. The mean of the feature is (\bar{X}). The overall amount of supplier logistics data samples is n. The supplier logistics data sample, standard mean, and standard deviation are used to calculate the Z score value.

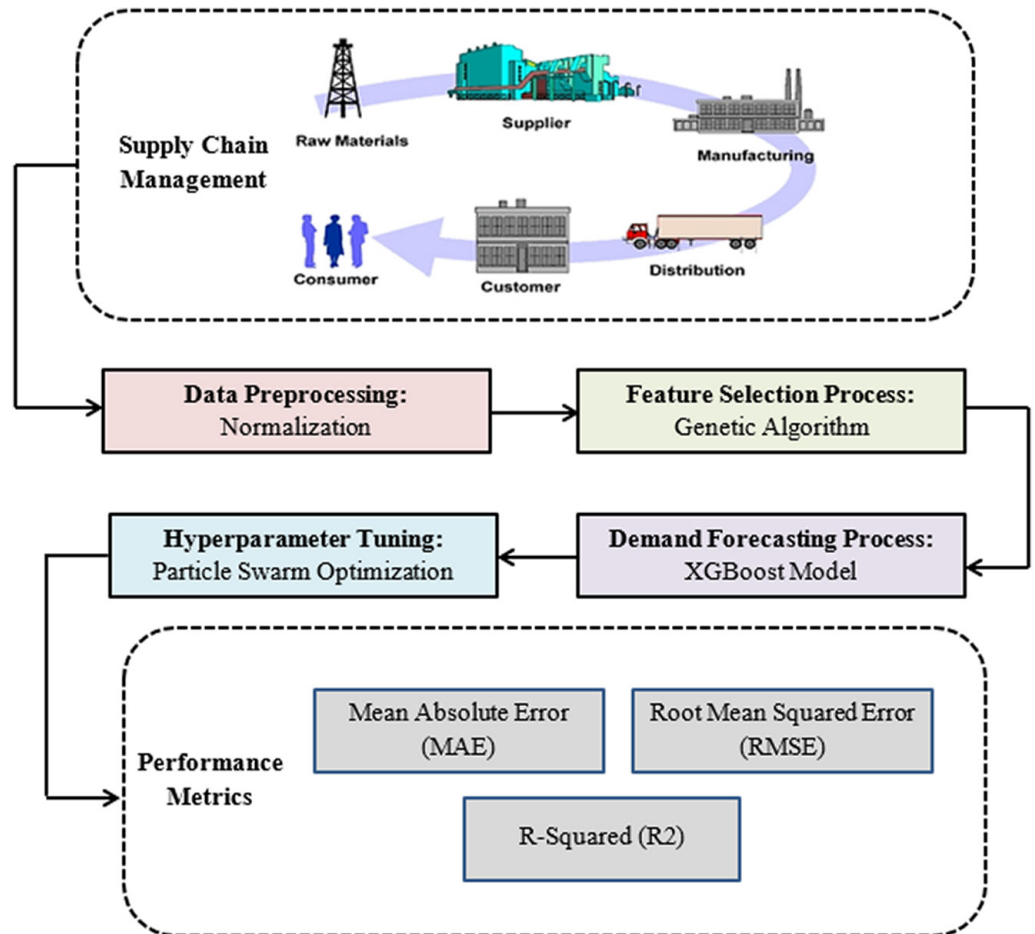


Fig. 1. Overall working process of the proposed method

3.2 Feature selection using GA

Next, GA is applied to FS process to identify the most informative variables. John Holland and his colleagues in 1975 initially coined the intelligent GA, an evolutionary algorithm. It is one of the most conventional search methods for resolving optimization problems based on Darwin's theory. It is a direct, derivative-free search technique that finds approximate or true solutions to search and optimization problems. The process of GA starts with the population initialization of n chromosomes within the range. Next, a fitness function (FF) is used to assess the fitness of individual chromosomes in the population. Following, GA operators (crossover plus mutation) are activated to generate the new population. This process is iteratively repeated until the ending condition is satisfied. An interconnection of GA phases is given in Figure 2.

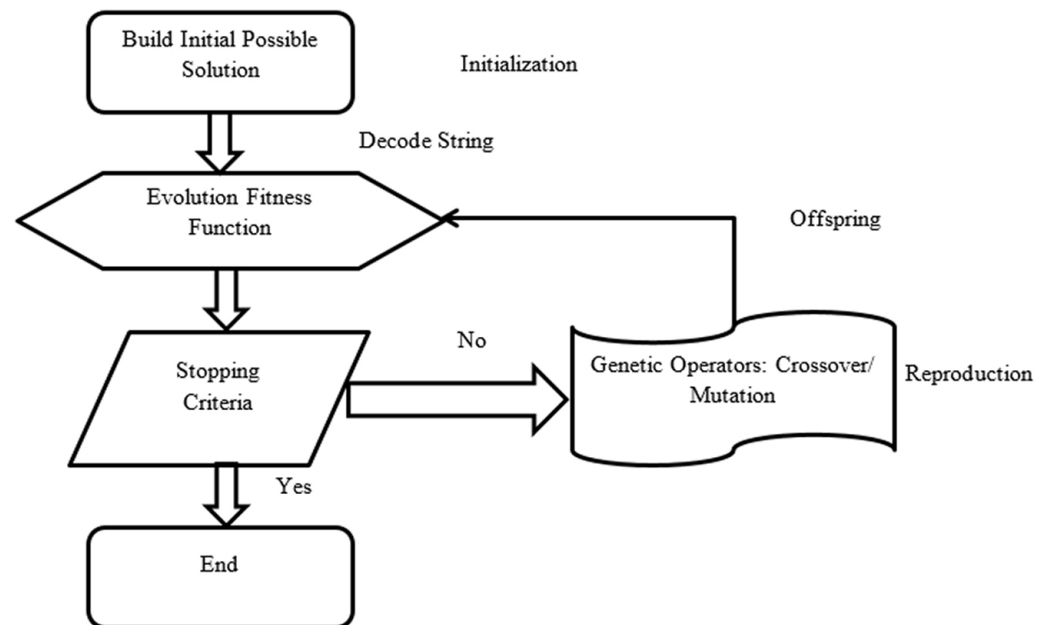


Fig. 2. Interconnection of GA phases

Steps involved in GA:

1. Create initial population of random solutions.
2. Assessment of the FF of every single solution in the population.
3. Generate new population by repeating the succeeding steps as follows:
 - Selection:* Choose a set of parent solutions from the population based on their fitness and higher fitness values are more likely to be selected.
 - Crossover:* Implement crossover using a predetermined crossover probability to generate new child solution.
 - Mutation:* Implement mutation using a predetermined mutation probability.
4. Adopt new population to further process the algorithm.
5. Check whether the ending condition is met. If yes, then terminate; or else return to step 2.

Single objective and multi-objective, binary and real encoded; unconstrained and constrained are different variants of GA, and the suitable variant might be applied based on the problem that has been dealt with.

3.3 XGBoost-based prediction

In this work, XGBoost model is employed for demand forecasting demonstrating the model’s capability to accurately predict future market trends. XGBoost is a gradient-boosting algorithm that is known for its speed and accuracy, making it suitable for the demand forecasting task. XGBoost optimizes forecast efficiency by capturing non-linear relationships in the data. XGBoost is a modular tree-boosting ML algorithm. The XGBoost scalability in each situation is the most critical factor for its success. The structure of XGBoost model is shown in Figure 3.

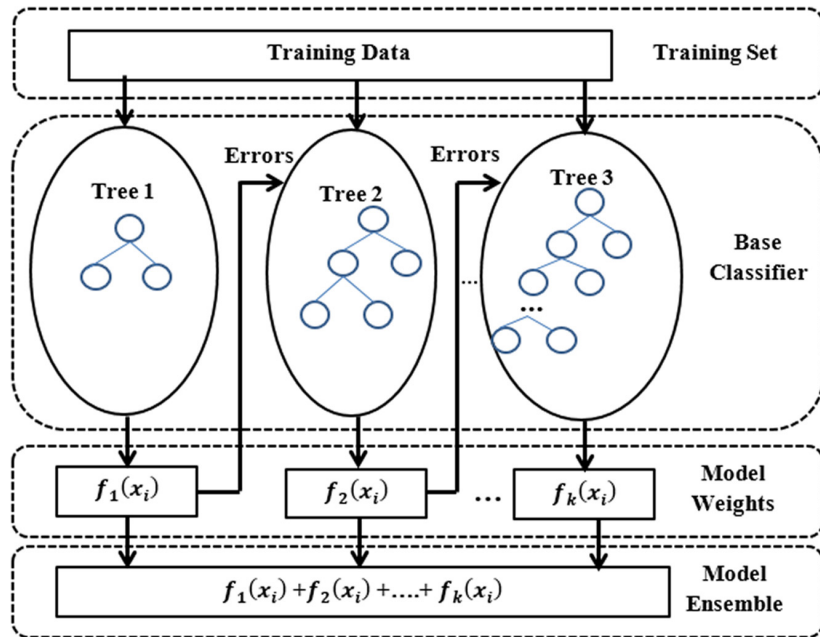


Fig. 3. Structure of XGBoost model

Incorporating various base classifiers (DTs) considerably improves prediction and classification performance. At first, the training dataset constructs an initial DTs, providing an initial prediction of the target variable and producing the output. In the following iteration, the algorithm analyzes the residuals between the actual values and the current predictions, and these residuals are passed into the succeeding DT. The newly generated tree is directed to learn and correct the deficiency of the preceding prediction. Every new DT aimed at minimizing the existing errors, enabling them to enhance the prediction iteratively. As soon as each DT is trained, their outcomes are combined using a weighted aggregation model to generate the final output. XGBoost successfully approximates the true target value, delivering accurate and highly effective predictions through the iteration process.

The XGBoost model used DTs as a base learner to create multiple weak learners. Then, it trains the model continuously towards gradient descent. Assume the model has k (DTs) as given in the below formula:

$$y_i = \sum_{t=1}^k f_t(x_i), f_t \in F \tag{3}$$

$$F = \left\{ f_t(x_i) = \omega_{q_t(x_i)} \right\} \tag{4}$$

Now, the number of trees can be denoted as k , the function in function space F is represented by f_p , the predicted value of the model is indicated as y_p , the input of i th data point is x_i , the weight of the leaf node where sample x_i occurs can be defined as $\omega_{q(x_i)}$ and the leaf node corresponding to sample x_i is indicated as $q(X)$. The XGBoost adopts an addition and forward distribution model, where all the iterations don't affect the original model,

$$\begin{cases} y_i^{(0)} = 0 \\ y_i^{(1)} = y_i^{(0)} + f_1(x_i) \\ \vdots \\ y_i^{(t)} = y_i^{(t-1)} + f_t(x_i) \end{cases} \tag{5}$$

The objective function (L) of XGBoost model has two components: the regularization term, which mitigates over-fitting and manages model complexity, and the loss function, which quantifies the error between the actual and predicted loads. The objective function can be mathematically represented as follows:

$$L = \sum_{i=1}^n l(y_i, y_i) + \sum_{j=1}^k \Omega(f_j) \tag{6}$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \tag{7}$$

Now, the error between the actual and predicted values is $l(y_i, \hat{y})$, the regularization term used to reduce over-fitting can be referred to as $W(f_j)$, the number of leaf nodes is T , the penalty factor is $T\gamma$, the weight of the leaf nodes is w , and the sample size is n .

$$L^{(t)} = \sum_{i=1}^n y_i \tag{8}$$

Since the preceding $t - 1$ regression trees are known information, viz., $W(f_j)$ is constant, the optimization method for the t th tree is unaffected. Hence, the objective function is expressed as

$$L^{(t)} = \sum_{i=1}^n l(y_i, y_i) + \Omega(f_t) = \sum_{i=1}^n l(y_i, y_i) + \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \tag{9}$$

In terms of leaf node traversal, the objective function can be rewritten by

$$L^{(t)} = \gamma T + \sum_{j=1}^T \left(\sum_{i \in I_j} l(y_i, y_i) + \frac{1}{2} \lambda \omega_j^2 \right) = \gamma T + \sum_{j=1}^T \left(\sum_{i \in I_j} l(y_i, y_i^{(t-1)} + \omega_j) + \frac{1}{2} \lambda \omega_j^2 \right) \tag{10}$$

The Taylor series expands the objective function for a binary function, then $l(y_i, y_i^{(t-1)} + \omega_j)$ is formulated by

$$l(y_i, y_i^{(t-1)} + \omega_j) = l(y_i, y_i^{(t-1)}) + l'(y_i, y_i^{(t-1)})\omega_j + \frac{1}{2} l''(y_i, y_i^{(t-1)})\omega_j^2 \tag{11}$$

By removing the term $l(y_i, y_i^{(t-1)})$, the objective function is simplified as:

$$\tilde{L}(t) = \gamma T + \sum_{j=1}^T \left(\sum_{i \in I_j} (h_i \omega_j + \frac{1}{2} g_i \omega_j^2) + \frac{1}{2} \lambda \omega_j^2 \right) = \gamma T + \sum_{j=1}^T \left(\omega_j \sum_{i \in I_j} h_i + \frac{1}{2} \omega_j^2 \left(\lambda + \sum_{i \in I_j} g_i \right) \right) \tag{12}$$

where $h_i = l'(y_i, y_i^{(t-1)})$ and $g_i = l''(y_i, y_i^{(t-1)})$

A lesser value of objective function shows the best regression tree structure. Minimalizing it and setting its derivative to 0 produces the weights for each leaf node.

The minimum loss ($\tilde{L}(t)$) can be derived by substituting them into the objective function,

$$\tilde{L}(q) = -\frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{i \in I_j} h_i \right)}{\sum_{i \in I_j} g_i + \lambda} + \gamma T \tag{13}$$

The architecture of XGBoost prediction model includes the following stages: (1) begin with a first iteration, sequentially building sub-models; (2) calculate the g_i and h_i first-and-second-order derivatives of the loss function before each iteration; (3) create new DTs and compute the predicted value for all the leaf nodes; and (4) after each iteration, add the new model to the existing model, creating the final prediction model through repetitive iterations. Businesses can leverage the power of state-of-the-art ML methods to make accurate predictions about future demand through the XGBoost model for demand forecasting. This aids them improve their bottom line, and their operations, and improve reduce costs.

3.4 PSO-based hyperparameter tuning

Finally, PSO algorithm is employed as hyperparameter tuning which allows the model to achieve optimal forecasting results. The PSO technique is a population-based stochastic search technique where the location of all the particles signifies a promising solution to the problem to be optimized, and the particle position is calculated by the fitness value based on the optimization function. The population can be randomly initialized as a set of random candidate solutions in the PSO, and then all the particles move within the given search range. After cyclic iterations, the optimum solution is attained until the stopping criteria are met.

PSO is derived from the behavior of bird flocking or fish schooling. Its main objective is to search for close-to-optimal or exact solutions to NP-hard problems in a search range by simulating the movement of particles. Each particle represents a potential solution, and its location and velocity are iteratively adjusted. Every i th particle, in a D -dimensional space, is characterized by its location, x_i , and velocity, v_i .

$$x_j = [x_{i1}, x_{i2}, \dots, x_{iD}] \tag{14}$$

$$v_j = [u_{i1}, u_{i2}, \dots, v_{iD}] \tag{15}$$

Based on Eq. (16), the velocity is updated for i th particles:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_j - x_i^t) + c_2 r_2 (g - x_i^t) \tag{16}$$

Now, the updated velocity of i th particle at $t + 1$ iteration is denoted by, v_i^{t+1} , the inertia weight to control the influence of preceding velocity can be represented

as w , c_1 and c_2 are the acceleration coefficients, the uniform distribution random value within $[0, 1]$ are r_1 and r_2 , the optimum location discovered by i th particle is p_i , the global optimal location found by the swarm is g ; and the existing location of i th particles obtain at t th iteration is x_i^t .

The location of all the particles is updated by:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (17)$$

The algorithm ends once a pre-defined number of iterations is obtained, and the solution selected is the final value of x_j particle.

4 EVALUATION METRICS

In this section, the demand forecasting outcomes of XGBoost method are tested on a dissimilar dataset. We evaluated the performance of all the models using different performance metrics such as MAE, RMSE, and R2 scores when the data was processed through different models. In Table 1, MAE, RMSE, and R2 scores of XGBoost, SARIMA, DT, and RF models were compared. Also, the outcome assured that the XGBoost has obtained an overall superior performance compared to SARIMA, DT, and RF techniques.

Table 1. Comparative outcome of XGBoost with other existing approaches

Models	MAE	RMSE	R2
XGBoost	72.4	71.3	75.2
SARIMA	69.4	56.9	49.9
DT	63.2	67.8	53.7
RF	59.1	54.2	63.1

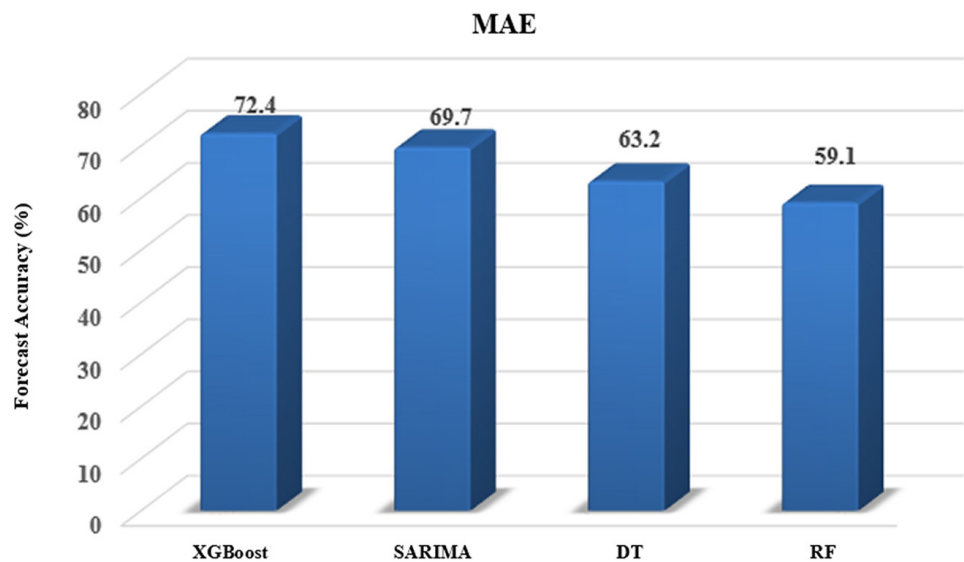


Fig. 4. Forecasting accuracy of the XGBoost models with other existing approaches

Figure 4 illustrates the forecasting accuracy of XGBoost model with other SARIMA, DT, and RF techniques. The experimental outcomes illustrate the superior

performance of the XGBoost algorithm, which accomplished the highest MAE values of 72.4%. Table 2, shows the MAE outcomes of XGBoost model with other techniques. In comparison, the SARIMA, DT, and RF algorithms demonstrated lower forecasting accuracy of 69.7%, 63.2%, and 59.1%. These findings highlight the proposed XGBoost model obtained maximum outcomes than the other SARIMA, DT, and RF techniques.

Table 2. MAE outcomes of XGBoost with other existing models

Models	MAE
XGBoost	72.4
SARIMA	69.4
DT	63.2
RF	59.1

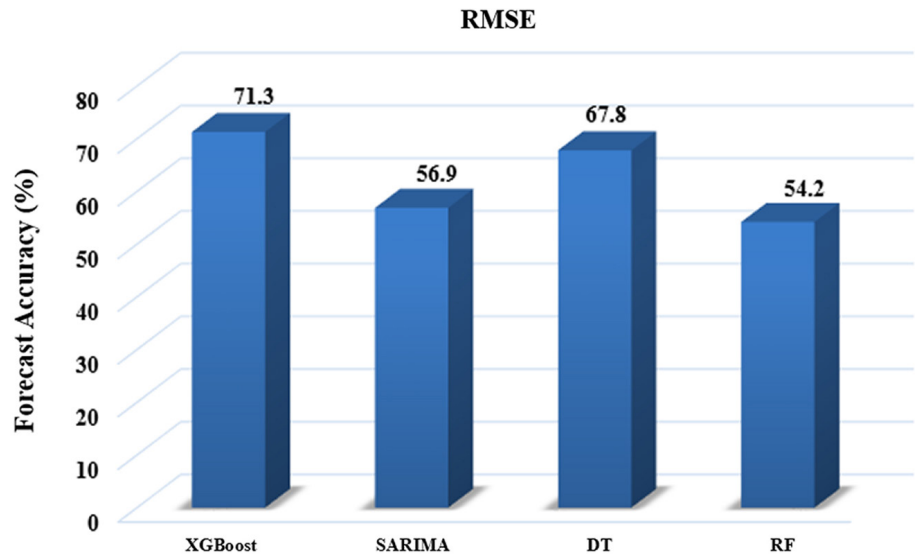


Fig. 5. Forecasting accuracy of the XGBoost models with other existing approaches

Figure 5 illustrates the forecasting accuracy of XGBoost model with other SARIMA, DT, and RF techniques. The experimental outcomes illustrate the superior performance of the XGBoost algorithm, which accomplished the highest RMSE values of 71.3%. Table 3, shows the RMSE analysis of XGBoost model with other existing techniques. In comparison, the SARIMA, DT, and RF algorithms demonstrated lower RMSE values of 56.9%, 67.8%, and 54.2%. These findings highlight the proposed XGBoost model obtained maximum forecasting outcomes than other SARIMA, DT, and RF techniques.

Table 3. RMSE outcomes of XGBoost with other existing models

Models	RMSE
XGBoost	71.3
SARIMA	56.9
DT	67.8
RF	54.2

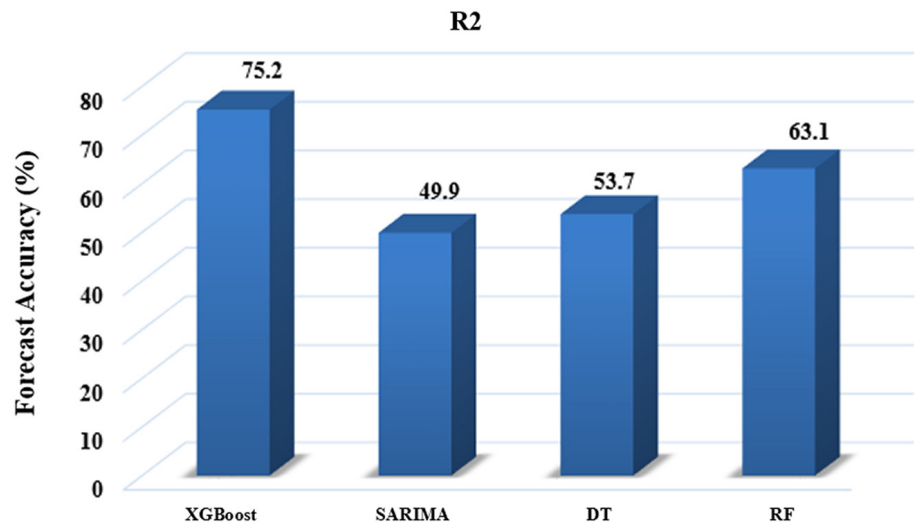


Fig. 6. Forecasting accuracy of the XGBoost models with other existing approaches

Figure 6 illustrates the forecasting accuracy of the XGBoost model with other SARIMA, DT, and RF techniques. The experimental outcomes illustrate the superior performance of the XGBoost algorithm, which obtained the highest R2 values of 75.2%. Table 4 shows the R2 analysis of the XGBoost model with other existing technique. In comparison, the SARIMA, DT, and RF algorithms demonstrated lower forecasting accuracy of 49.9%, 53.7%, and 63.1%. These findings highlight the proposed XGBoost model obtained maximum forecasting outcomes than the other SARIMA, DT, and RF techniques.

Table 4. R2 outcomes of XGBoost with other existing models

Models	R2
XGBoost	75.2
SARIMA	49.9
DT	53.7
RF	63.1

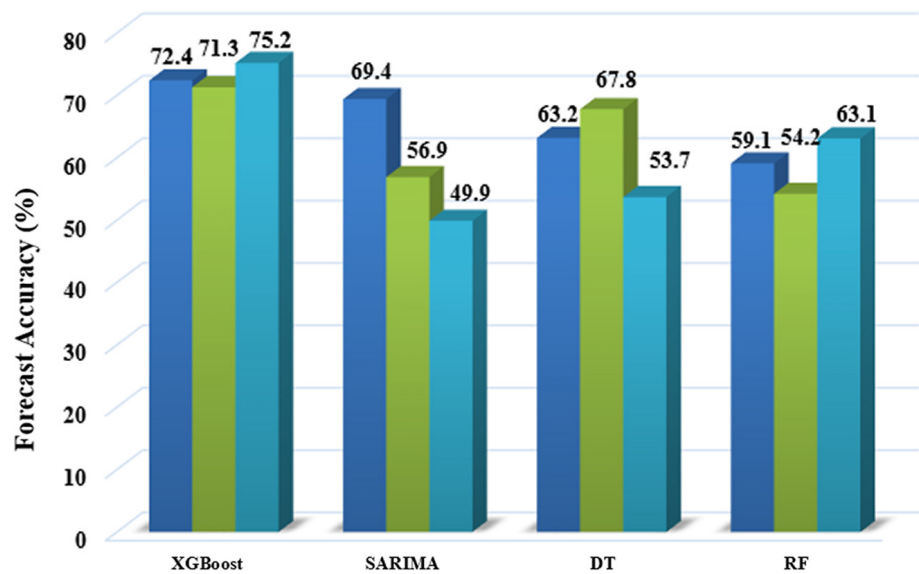


Fig. 7. Comparative analysis of the XGBoost models with other existing approaches

Figure 7 demonstrates the comparative analysis of the XGBoost model with SARIMA, DT, and RF techniques. In Table 1, a brief comparison analysis of the proposed method takes place. The experimental outcome demonstrates that the proposed EX model accomplishes outstanding forecasting performance than SARIMA, DT and RF techniques.

5 CONCLUSION

In this paper, we introduce XGBoost-based demand forecasting in SCM using an ML algorithm. At the initial stage, the Z-score normalization serves as a data pre-processing to normalize the input data features into uniform scales. For feature selection process, the GA is exploited to identify the most descriptive variables. Then, the XGBoost model is employed for the demand forecasting process to accurately forecast future market trends. Lastly, PSO approach involves hyperparameter tuning which allows the model to achieve optimum forecasting outcomes. Experimental outcomes highlight that the proposed technique achieves higher forecasting efficiency than other existing methodologies.

6 REFERENCES

- [1] H. Kagalwala, G. V. Radhakrishnan, I. A. Mohammed, R. R. Kothinti, and N. Kulkarni, "Predictive analytics in supply chain management: The role of AI and machine learning in demand forecasting," *Advances in Consumer Research*, vol. 2, pp. 142–149, 2025.
- [2] J. Zhong, X. Hu, O. A. Alghamdi, S. Elattar, and S. Al Sulaie, "XGBoost with Q-learning for complex data processing in business logistics management," *Information Processing & Management*, vol. 60, no. 5, p. 103466, 2023. <https://doi.org/10.1016/j.ipm.2023.103466>
- [3] Z. N. Jawad and B. Villányi, "Designing predictive analytics frameworks for supply chain quality management: A machine learning approach to defect rate optimization," *Platforms*, vol. 3, no. 2, p. 6, 2025. <https://doi.org/10.3390/platforms3020006>
- [4] A. R. Muthukalyani, "Unlocking accurate demand forecasting in retail supply chains with AI-driven predictive analytics," *Information Technology and Management*, vol. 14, no. 2, pp. 48–57, 2023.
- [5] U. Nweje and M. Taiwo, "Leveraging artificial intelligence for predictive supply chain management, focus on how AI-driven tools are revolutionizing demand forecasting and inventory optimization," *International Journal of Science and Research Archive*, vol. 14, no. 1, pp. 230–250, 2025. <https://doi.org/10.30574/ijrsra.2025.14.1.0027>
- [6] B. Aliahmadi, "Design and development of a machine learning-based supply chain optimization model: A framework for multiobjective smart decision making," *Journal of Business and Future Economy*, vol. 1, no. 3, pp. 1–13, 2024.
- [7] P. Verma, "Transforming supply chains through AI: Demand forecasting, inventory management, and dynamic optimization," *Integrated Journal of Science and Technology*, vol. 1, no. 9, 2024.
- [8] S. F. B. S. Zain, "Evaluating the impact of machine learning-driven supply chain digital twins on sustainability, agility, and operational performance," *International Journal of Applied Business Intelligence*, vol. 2, no. 12, pp. 7–13, 2022.
- [9] T. Lu, E. Garcia, and J. Lee, "Optimizing supply chain demand forecasting and inventory management using large language models," *International Journal of Innovative Research in Computer Science and Technology*, vol. 12, no. 6, pp. 89–94, 2024. <https://doi.org/10.55524/ijrcst.2024.12.6.12>

- [10] M. P. R. Mahin, M. Shahriar, R. R. Das, A. Roy, and A. W. Reza, "Enhancing sustainable supply chain forecasting using machine learning for sales prediction," *Procedia Computer Science*, vol. 252, pp. 470–479, 2025. <https://doi.org/10.1016/j.procs.2025.01.006>
- [11] R. M. Prabhu, G. Hema, S. Chepure, and M. N. Guptha, "Logistics optimization in supply chain management using clustering algorithms," *Scalable Computing: Practice and Experience*, vol. 21, no. 1, pp. 107–114, 2020. <https://doi.org/10.12694/scpe.v21i1.1628>
- [12] S. K. Jauhar and M. Pant, "Genetic algorithms in supply chain management: A critical analysis of the literature," *Sādhanā*, vol. 41, pp. 993–1017, 2016. <https://doi.org/10.1007/s12046-016-0538-z>
- [13] S. Zeng, C. Liu, H. Zhang, B. Zhang, and Y. Zhao, "Short-term load forecasting in power systems based on the Prophet-BO-XGBoost model," *Energies*, vol. 18, no. 2, p. 227, 2025. <https://doi.org/10.3390/en18020227>
- [14] S. Taghiyeh, D. C. Lengacher, A. H. Sadeghi, A. Sahebi-Fakhrabad, and R. B. Handfield, "A novel multi-phase hierarchical forecasting approach with machine learning in supply chain management," *Supply Chain Analytics*, vol. 3, p. 100032, 2023. <https://doi.org/10.1016/j.sca.2023.100032>
- [15] J. Liu, "Strategy and practice for improving supply chain management through mobile interaction technology," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 19, no. 4, pp. 178–192, 2025. <https://doi.org/10.3991/ijim.v19i04.54219>

7 AUTHORS

Dr. S. Nagadevi, who has over 20 years of teaching experience, earned a B.Tech in Computer Science and Engineering from SRM Engineering College, Chennai, Tamil Nadu, is a member of Madras University. In 2006, she graduated with an M.E. in computer science and engineering from Tagore Engineering College, a Tamil Nadu institution affiliated with Anna University. She is working as an Assistant Professor in the Department of Computer Science and Engineering at SRMIST in Chennai. She completed her research in the area of cloud computing. She has authored on multiple research papers on topics such as virtual machine placement, cloud data centers, and cloud computing, which has got published in reputable journals including Lecture Notes in Networks and Systems, and IOS Press. She has framed many courses in the B.Tech level including GPU Programming, GPU Powered Computing, Sparse intensive GPU Computing, Parallel Programming with OpenMP and MPI and so on. She is popularly known for her Work in the area of GPU Computing. As a mentor for the SRM-ASI student chapter branch, she has collaborated with IIM Bangalore (E-mail: nagadevs@srmist.edu.in).

Dr. G. Abirami is an Assistant Professor in the Department of Computing Technologies at the School of Computing, SRM Institute of Science and Technology, Kattankulathur. With over 14+ years of teaching experience at SRMIST, she completed her PhD in Computer Science and Engineering with a thesis titled "Attribute Based Access Control Using Unified Dynamic Trust Metrics in the Application of Business Data" in October 2021. Her current research interests focus on object detection using deep learning technique and healthcare image processing (E-mail: abiramig@srmist.edu.in).

Dr. R. Vidhya is working as an Associate Professor in the Department of Computing Technologies at SRM Institute of Science and Technology, Chennai, INDIA. She earned her B.E. in CSE in 2005, her M.E. in CSE in 2011, and her PhD in Computer Science and Engineering from SRM Institute of Science and Technology in 2021 respectively. She graduated from SRMIST in 2021 with an MSc in Yoga for

Human Excellence based on interest. She has over 18 years of experience, and has produced numerous research papers, projects, and held leadership positions on a number of university committees. She has participated in a number of national or international symposium or workshop program committees. She is currently interested in learning analytics, Big Data, Data mining and Machine Learning (E-mail: vidhyar@srmist.edu.in).

Dr. S. Selvakumar is currently working as Professor in the Department of Computer Science and Business, Panimalar Engineering College in Chennai, India. He holds a Ph.D. in Computer Science and Engineering from Karpagam Academy of Higher Education, Coimbatore, India. He has 18 years of teaching experience in reputed engineering colleges and university. With a career dedicated to teaching and research, his expertise spans Machine Learning, Deep Learning, Computer Network, Cloud Computing and Data Analytics. He has been granted one design patent, has published two patents, two books and has authored more than 60 research papers in reputed journals and conferences (E-mail: selvathendral1981@gmail.com).

Dr. C. Vijayalakshmi is currently working as an Assistant Professor (Sr. Grade) in the Department of Computer Science and Engineering at the School of Computer Information, and Mathematical Sciences, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She holds a Ph.D. in Information Technology from B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She has 21 years of teaching experience in reputed engineering colleges and university. With a career dedicated to teaching and research, her expertise spans Machine Learning, Deep Learning, and Data Analytics. She has been granted one patent, has published six patents, and has authored more than 20 research papers in reputed journals and conferences (E-mail: vijayalakshmi@crescent.education).