TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 24, No. 6, December 2026, pp. 2255~2272
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i6.27715

Data-driven machine learning-based demand forecasting for supply chain management

Norah Nasser M ALQutaim, Abdulrhman ALShareef, Syed Hamid Hasan, Usman Ali Khan
Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Article Info

Article history:

Received Dec 13, 2025
Revised Oct 4, 2026
Accepted Nov 15, 2026

Keywords:

Demand forecasting
Gradient boosted decision tree
Inventory optimization
Logistic regression tree
Random forest
Stacked ensemble
Supply chain management

ABSTRACT

Demand forecasting is crucial in supply chain management (SCM), especially in the fashion and beauty industry, where demand variability and the complexity of supply chain operations can impact inventory and logistics decisions. In this study, a stacked ensemble-based random gradient logistic regression (SE-RGLR) model is proposed for accurate demand forecasting and supply chain decision making. The proposed approach employs random forest (RF), gradient boosting and logistic regression (LR) in a stack ensemble setting to learn complementary patterns from sales, product, supplier and logistic data. The dataset is preprocessed by handling missing values, removing duplicates, transforming features, handling outliers and partitioning the training-testing data. The results of the experiments show that the proposed SE-RGLR model achieves 98.6% accuracy, 97.5% precision, 99.4% recall, 98.5% specificity and 99.2% F1-score. The values of mean absolute error (MAE), mean squared error (MSE) and root MSE (RMSE) are 0.78, 0.79, and 0.82 respectively. The results indicate the applicability of the proposed ensemble framework to support demand forecasting, inventory planning, supplier evaluation and logistics decision-making in SCM.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Usman Ali Khan
Department of Information Systems, Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, Saudi Arabia
Email: ukhan@kau.edu.sa

1. INTRODUCTION

Demand forecasting is a key element of supply chain decision-making and is critical to the overall performance of supply chain systems [1]. It aims to anticipate future customer requirements by leveraging historical sales data and recent market trends, particularly in fast-changing industries such as fashion and beauty [2]. Accurate demand forecasting supports cost reduction and enables improved service delivery by assisting organizations in inventory control, production planning, and logistics coordination [3].

However, conventional demand forecasting approaches struggle to cope with the increasing complexity and uncertainty of modern supply chain environments. Traditional statistical models often fail to handle multi-source data integration, pronounced demand seasonality, and sensitivity to external shocks such as promotional campaigns, supplier disruptions, and global events [4]. These limitations reduce their effectiveness in industries characterized by volatile demand and short product life cycles.

The increasing complexity and unpredictability of modern supply chain environments necessitate the adoption of advanced data analytics and machine learning techniques for effective supply chain management (SCM). Modern businesses rely heavily on efficient SCM systems to remain competitive in the

presence of fluctuating demand, global disruptions, and the growing need for real-time decision-making. Inefficiencies in forecasting and planning often lead to high operational costs, inventory shortages, delayed deliveries, and reduced supplier performance.

Reducing uncertainty, improving decision-making accuracy, and achieving sustainable growth while maintaining profitability and operational resilience are key objectives of contemporary SCM. Consequently, the adoption of intelligent forecasting approaches capable of adapting to complex, dynamic, and data-rich environments has become essential. These challenges underscore the need for advanced ensemble-based machine learning models that can enhance forecasting accuracy and support responsive, cost-effective supply chain operations.

In contrast, machine learning-based forecasting techniques enable data-driven decision-making by learning complex nonlinear patterns from large-scale historical and real-time data [5]. Such data may include pricing strategies, promotional activities, weather conditions, inventory levels, and evolving market trends [6]. By dynamically adapting to changing conditions, machine learning models transform supply chains from reactive systems into proactive and predictive frameworks, reducing the risks of overstocking and stockouts while improving operational efficiency and managerial decision-making [7], [8]. Recent studies have demonstrated the effectiveness of ensemble and hybrid learning models for intelligent supply chain forecasting and optimization in Internet of Things (IoT)-enabled and cloud-based environments [9]–[11].

Despite the growing adoption of machine learning for demand forecasting, most existing studies focus on single-source datasets or address isolated forecasting tasks. Moreover, limited attention has been given to integrated ensemble frameworks that simultaneously support demand forecasting, inventory management, supplier evaluation, and logistics optimization in fashion and beauty supply chains. This research gap motivates the proposed stacked ensemble-based random gradient logistic regression (SE-RGLR) approach, which aims to provide a comprehensive and robust forecasting solution for complex supply chain environments.

This study proposes a novel SE-RGLR framework that systematically integrates random forest (RF), gradient boosting decision trees (GBDT), and logistic regression (LR) within a single ensemble architecture. Unlike existing studies that rely on individual or loosely coupled models, the proposed framework exploits model diversity through stacking to improve robustness and predictive accuracy under uncertain and dynamic supply chain conditions. In contrast to prior works that address demand forecasting, inventory management, supplier evaluation, or logistics optimization as isolated problems, the proposed SE-RGLR framework provides a unified solution capable of supporting multiple supply chain decision domains simultaneously.

The proposed approach is designed to effectively integrate heterogeneous supply chain data, including sales trends, inventory levels, supplier performance indicators, and logistics attributes, thereby overcoming the limitations of single-source data-driven forecasting models commonly reported in the literature. The framework is specifically tailored to the fashion and beauty industry, characterized by high demand volatility, short product life cycles, and complex logistics structures, demonstrating the applicability of ensemble machine learning techniques to real-world, industry-specific supply chain challenges.

The main contributions of this work are as follows:

- Development of a novel SE-RGLR stacked ensemble framework integrating RF, GBDT, and LR.
- Improved demand forecasting accuracy through ensemble learning.
- Integrated optimization of inventory, supplier performance, and logistics decisions.
- Demonstration of practical applicability in the fashion and beauty supply chain domain.
- Engineering-oriented intelligent decision support: the proposed framework integrates heterogeneous data processing and ensemble learning for adaptive and computationally efficient decision-making in IoT-enabled and intelligent engineering systems.

The proposed SE-RGLR framework, from a multidisciplinary engineering perspective, combines ensemble machine learning with predictive analytics in order to enable intelligent decision-making in electrical, control, telecommunication and computing applications, providing adaptive prediction and computationally-efficient data processing. In particular, its integration with IoT-enabled inventory monitoring systems supports real-time data acquisition and control-oriented decision-making. The framework can be deployed over cloud-based and edge computing infrastructures, enabling scalable and intelligent supply chain analytics through reliable telecommunication networks. Furthermore, the lightweight ensemble learning design aligns with intelligent computing systems that require efficient, data-driven optimization under constrained computational environments.

Beyond the present application, SE-RGLR is designed as a generalizable ensemble framework that can be adapted to a range of engineering domains where heterogeneous data, predictive forecasting, and intelligent decision support are required. Its underlying architecture is not restricted to inventory monitoring and can be extended to applications such as smart grids, intelligent transportation systems, industrial

automation, and other cyber-physical engineering environments. This domain-independent structure provides a foundation for developing adaptive, data-driven engineering solutions while allowing domain-specific features, constraints, and operational requirements to be incorporated without altering the core ensemble learning strategy.

The organization of this research is structured as follows: section 2 contains various literature works with drawbacks and research gaps. The proposed methodology is clearly described in section 3. The experimental validation and its graphical representations are provided in section 4. Finally, this paper is concluded in section 5 with future work.

2. RELATED WORK

Several studies have explored advanced analytical and machine learning techniques to improve forecasting and optimization in SCM. Existing approaches can be broadly categorized into optimization-based methods, deep learning-based forecasting models, and hybrid or ensemble-driven intelligent systems. Optimization-based approaches primarily focus on minimizing operational costs and improving resource utilization in complex supply chain networks. For instance, Shokouhifar *et al.* [12] proposed a hybrid whale optimization algorithm combined with variable neighborhood search to reduce economic costs in real-world supply chain scenarios. Similarly, Benmamoun *et al.* [13] introduced the wombat optimization algorithm to address the challenges of dynamic and large-scale supply chain networks, demonstrating superior performance compared to traditional metaheuristics. Pan *et al.* [14] addressed multi-echelon and multi-period supply chain optimization through a perishable product supply chain network (PPSCN) model, highlighting the effectiveness of integrated location-inventory-routing solutions.

Deep learning-based forecasting models have gained prominence due to their ability to capture nonlinear relationships and temporal dependencies in demand patterns. Aldahmani *et al.* [15] proposed the uni-regression deep approximate forecasting (UDAF) model to handle dynamic demand influenced by external factors. Wang *et al.* [16] combined long short-term memory (LSTM) with autoregressive moving average (ARMA) to improve prediction accuracy through error compensation mechanisms. Further improvements in deep learning performance have been achieved through optimization-enhanced models such as Jaya-optimized LSTM (JLSTM) [17] and Bayesian error reduction LSTM (BER-LSTM) [18], where metaheuristic algorithms are employed to fine-tune hyperparameters, leading to reduced prediction errors.

In parallel, several data-driven and intelligent frameworks have been developed for related predictive and optimization tasks. Abdullah *et al.* [19] utilized a soft gated recurrent unit (SGRU) for traffic congestion prediction in smart cities, while Chetty *et al.* [20] proposed a tree-based deep convolutional neural network optimized using sheep flock optimization for workload and energy prediction in cloud data centers. Although these approaches demonstrate strong predictive capabilities, most are designed for domain-specific applications and do not directly address integrated supply chain decision-making. Aljabhan and Obaidat [21] developed the perceptive craving game search (PCGS) approach to generate optimal keys for data sanitization. In their method, data was sanitized at the manufacturer's end, safely transferred via blockchain sub-chains, and restored at the customer's end by applying the same key. The evaluation result exhibited that the developed framework attained better key generation, efficiently protecting data sanitization as opposed to unauthorized access.

In addition to machine learning-based forecasting and optimization, signal processing, embedded systems, and intelligent control frameworks play an important role in developing intelligent engineering systems. Signal processing techniques support the extraction, filtering, and transformation of time-dependent and noisy data, which can improve the quality of inputs used for predictive modeling [22]. Embedded computing systems enable localized data processing and real-time monitoring in resource-constrained environments, particularly in IoT-enabled industrial applications [23]. Furthermore, intelligent control frameworks integrate predictive information with decision-making and control mechanisms to improve system responsiveness, stability, and operational efficiency [24]. The integration of machine learning with these engineering paradigms provides opportunities for developing adaptive and data-driven systems capable of supporting real-time supply chain monitoring, inventory control, and logistics coordination.

2.1. Limitations and research gaps

The above-discussed research works reveal several limitations in current SCM and demand forecasting approaches. One key limitation is the limited interpretability of many machine learning models, which makes it difficult to explain prediction outcomes and build trust among decision-makers. In addition, integrating advanced machine learning techniques into existing SCM systems remains challenging, as it requires expertise across both operational and analytical domains.

More importantly, most existing studies focus on single-model or single-source data-driven approaches, despite the fact that modern supply chains generate heterogeneous data from multiple sources

such as sales transactions, inventory systems, supplier records, IoT-enabled sensors, market trends, and external environmental factors. Furthermore, many approaches address individual supply chain functions in isolation, such as demand forecasting or inventory optimization, without providing a unified framework for multi-domain decision-making. These limitations restrict the ability of current methods to effectively handle demand seasonality, external shocks, and complex interdependencies across supply chain processes.

Consequently, there is a clear research gap for integrated stacked ensemble machine learning frameworks that can combine multiple learning models and leverage multi-source data to simultaneously support demand forecasting, inventory management, supplier performance evaluation, and logistics optimization. Addressing this gap forms the core motivation for the proposed SE-RGLR framework, which aims to provide a holistic and robust solution for supply chain optimization in dynamic and uncertain environments.

3. METHOD

Figure 1 depicts a model for optimizing supply chain operations using a data-driven approach. The main goal of this research is to enhance the efficiency and performance of the company's supply chain by utilizing data-driven insights and predictive analytics. The following diagram provides an overview of the various phases involved in this study.

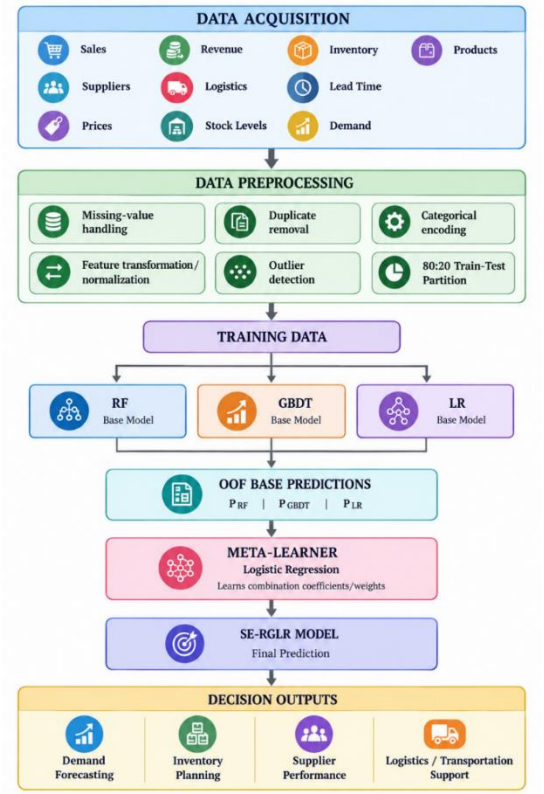


Figure 1. Proposed data-driven supply chain optimization framework illustrating the phases of data acquisition, preprocessing, and machine learning-based model development for multi-domain SCM tasks

- Phase 1: data acquisition
  During this phase, data is sourced from various departments within the fashion and beauty company, including supply chain records, product specifications, sales figures, revenue data, block levels, supplier information, and logistics details.
- Phase 2: data pre-processing
  In this phase, the collected data is cleaned and prepared. This process involves addressing missing data, eliminating duplicates, transforming features, detecting and handling outliers, and splitting the data into training and testing sets for model development.

– Phase 3: model development and optimization
  The final phase involves developing, training, and evaluating machine learning models using predefined hyperparameters and objective performance criteria. A range of models are applied, including RF, GBDT, LR, and stacked ensemble. These models are leveraged for different objectives, such as predicting demand, optimizing inventory, evaluating supplier performance, and improving logistics processes.

3.1. Data description

In this analysis, the applied dataset acquired data from the Kaggle database [25] and is designed for supply chain operations of the beauty and fashion profession specialized in cosmetics. To enhance the effectiveness of the supply chain, the dataset contains a wider array of attributes, collected over multiple time periods. It comprises necessary goods details including pricing, availability, and type which acquire insights into the strategies of pricing and goods limit. An evaluation of inventory management and sales efficacy enables operational data such as sales figures, income, and goods quantity. Travel times, shipping times, transporter, and expenses are certain logistical information that improves the transportation understanding and delivery procedure. Supplier data pinpoints reliability and quality control of the supplier such as lead times, manufacturing volumes, cost, and fault rates. The dataset also includes transportation modes, routes, and related costs authorizing a comprehensive analysis of logistics operations. All experiments were conducted using this publicly available dataset to ensure transparency and reproducibility of the proposed approach.

3.2. Data pre-processing

The tabular supply chain dataset is cleaned and prepared to ensure suitability for predictive modeling and analysis. Preprocessing steps include handling missing data, removing duplicates, feature transformation (normalization or standardization), outlier detection and handling, and partitioning the dataset into training and testing sets. These steps ensure reliable inputs for tasks such as demand prediction, inventory management, and supplier performance evaluation. Median imputation was adopted for handling missing values due to its robustness against outliers, while normalization was applied to numerical features to ensure comparable feature scales and stable model convergence. These preprocessing steps were consistently applied across all experiments to maintain reproducibility.

The procedures are:

a. Handling missing data: it is a vital phase in the pre-processing of the data that confirms the dataset's usability and integrity for analysis. Typical approaches include imputing missing values and eliminating rows or columns that have a significant number of missing data.

$$w_{imputed} = \frac{\sum_{j=1}^{m} w_j}{m} \quad (1)$$

b. Removing duplicates: this process is applied to eliminate data that are not necessary for a particular task. Eliminating duplicates can improve clarity, making it easier to extract meaningful insights.

c. Feature transformation: this step involves normalizing or standardizing the features to bring them to a similar scale, which is crucial for the performance of many machine learning algorithms.

$$w_{scaled} = \frac{w - w_{min}}{wmin_{max}} \quad (2)$$

where w is the feature value, w_{min} and w_{max} are the minimum and maximum values of the feature respectively.

d. Outlier detection and handling: this step focuses on detecting and managing data points that differ from the remaining dataset. Outliers can be addressed by either removing them or applying capping techniques.

$$Y = \frac{w_d - \lambda}{\sigma_d} \quad (3)$$

where w_d, λ, and σ_d signifies the individual data point, mean, and standard deviation. Outliers are typically identified when $|Y| > 3$.

e. Splitting data for model training and testing: this step entails partitioning the dataset into separate training and testing sets to enable objective evaluation of model generalization using standardized performance metrics. A fixed 80:20 training–testing split was employed consistently across all experiments to evaluate model performance.

$$Trainingsetsize = m \times train_size \quad (4)$$

where m is the number of data points, and $train_size$ is the proportion of data used for training (80%) and the rest 20% is for testing the data.

3.3. Model development

The SE-RGLR model is a powerful tool for optimizing key supply chain functions, including demand forecasting, inventory management, supplier performance, and logistics. By combining different machine learning techniques such as RF, GBDT, LR, and stacked ensembles, the SE-RGLR model integrates complementary learning mechanisms within a unified decision-making framework, as illustrated by Figure 2. RF and GBDT were selected for their ability to model nonlinear relationships and handle complex feature interactions, while LR provides interpretability and stable probabilistic outputs. The stacked ensemble combines these strengths to improve overall forecasting robustness. All base learners and the stacked ensemble model were implemented using standard machine learning libraries with consistent training–testing splits to ensure fair comparison and repeatability.

In this study, standard hyperparameter settings commonly reported in prior machine learning literature were adopted for all base models to ensure stable learning and fair comparison. Extensive hyperparameter tuning was not the primary focus of this work, as the objective was to evaluate the effectiveness of the proposed stacked ensemble framework rather than optimize individual model configurations. The experiments used fixed hyperparameter configurations for reproducible comparison rather than an extensive hyperparameter search. The principal configurations included 100 trees for RF, a maximum tree depth of 10, a learning rate of 0.1 for gradient boosting, and a regularization parameter \($C = 1.0$\) with L2 penalty for LR. Since RF, GBDT, and LR are non-iterative epoch-based models in the implemented configuration, training epochs were not applicable.

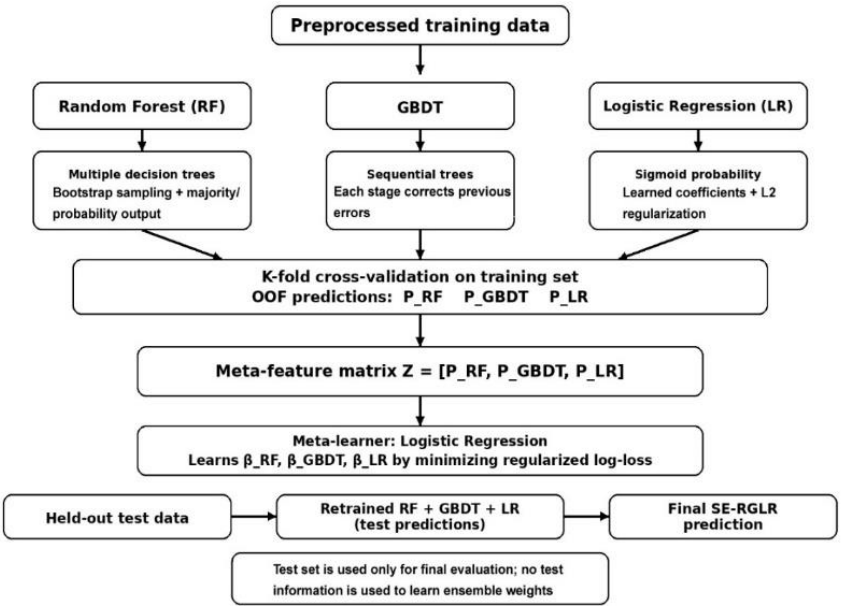


Figure 2. SE-RGLR stacking architecture and ensemble-weight learning

The following provides a detailed explanation of the SE-RGLR methodology. RF [26] constructs a collection of decision trees, where each tree is trained on a random subset of the data using bootstrapping. The final predictions are made by aggregating the individual tree outputs through a majority vote.

$$\hat{y}_{RF}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \quad (5)$$

where, B = number of decision trees, $T_b(x)$ = prediction generated by the b^{th}decision tree, x = input feature vector, and $\hat{y}_{RF}$ = aggregated RF prediction. For classification, the final class can alternatively be represented by majority voting: where $f_s(W)$ is the s^{th} decision tree model, and S is the total number of trees.

GBDT [27] constructs trees in a sequential manner, where each new tree attempts to correct the errors made by the previous one. The goal is to minimize the loss function, as shown in (6).

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (6)$$

where:
- $F_m(x)$ is the ensemble prediction after the m^{th} boosting stage
- $h_m(x)$ is the newly trained decision tree
- η is the learning rate
- m is the boosting iteration

The objective can be expressed as:

$$\mathcal{L}_{GBDT} = \sum_{i=1}^{N} L\left(y_i, F_M(x_i)\right) + \Omega(F)$$

where L represents the prediction loss and $\Omega(F)$ represents the model-complexity/regularization component where applicable.

LRT [28] divides the nodes based on more complex decision criteria, rather than using simple rules. The output at each leaf node represents the probability of the data point belonging to a specific class.

$$O(z|W) = \frac{1}{1+e^{-\left(x_0+\sum_{j=1}^{l} x_j w_j\right)}} \quad (7)$$

where x_j are the learned weights from the LR.

The stacked ensemble [29] model is structured to identify the most effective way to merge the results from individual base models. It feeds the predictions from these models into a meta-model, which then synthesizes them to generate the final outcome. Let the predictions from the base models be $\hat{z}_{RF}$, $\hat{z}_{GBDT}$, and $\hat{z}_{LRT}$. The stacked model learns a function e_{stack} to combine these predictions.

$$\hat{z}_{stacked} = x_1\hat{z}_{RF} + x_2\hat{z}_{GBDT} + x_3\hat{z}_{LRT} \quad (8)$$

where x_1, x_2, and x_3 are the learned weights of the base model predictions, typically learned through training the meta-model. After training the base models, their predictions are treated as input features for the stacked ensemble model. The ensemble model then learns the optimal weights for combining these outputs, ultimately generating the final prediction. In implementation, this combination is realized by training a meta-level learner on the predictions of the base models rather than on the original input features. The ensemble weights are therefore learned by the meta-model from the predictions generated by the RF, GBDT, and LR base learners. The learned combination is then applied to the unseen test data, while the fixed 80:20 training-testing protocol is maintained consistently across the comparative models for validation.

The objective function for training the stacked ensemble model can be expressed as the following loss function. The loss function measures the difference between the final prediction generated by the stacked ensemble and the corresponding true target value. During meta-model training, the loss is minimized to determine the model parameters that provide the most suitable combination of the base-model predictions.

$$K_{stack} = \sum_{j=1}^{m}\left(z_j - \hat{z}_{stacked}(W_j)\right)^2 \quad (9)$$

$\hat{z}_{stacked}(W_j)$ is the prediction from the stacked ensemble, and z_j is the true value.

In practical implementation, the SE-RGLR framework is executed in three sequential stages. First, the base learners (RF, GBDT, and LR) are independently trained using the preprocessed training dataset. Second, predictions generated by these base models on the training data are collected and used as input features for training the meta-model. Finally, the trained meta-model combines the base learner outputs to produce the final prediction on unseen test data. This implementation directly corresponds to the mathematical formulations presented in (5)-(9).

The implementation was conducted using Python 3.9 in Jupyter Notebook on a system equipped with an Intel Core i7 processor, 16 GB RAM, and a 512 GB SSD. Model performance was evaluated on a held-out test set using standardized classification and error-based metrics, as described in the results section.

3.3.1. Accurate demand forecasting in SCM with SE-RGLR

It is the forecasting demand [30] that assists businesses at various stages of management to plan their operation according to the values placed by their consumers. The SE-RGLR model improves this

process by combining different machine learning methods including RF, BGDT, and LR trees into an integrated framework which in this case is an ensemble framework. Moreover, the SE-RGLR model incorporates trend-related factors such as sales history, market indicators, and inventory status into the forecasting process. The outputs of multiple base models are summed in this model and this assists the model to extend its forecasting approach progressively. In this respect, the SE-RGLR model updates its forecasted values as it obtains the results and feedback from the environment. The forecasting framework is designed to update predictions as new data and market trends become available. Consequently, the forecasting outputs can support inventory planning decisions related to stock availability and replenishment. For demand prediction, the SE-RGLR model estimates the relationship between input features and the target demand variable using ensemble learning. The general process can be formulated as (10). Let us consider, $W = [w_1, w_2, \ldots w_m]$, $z = [z_1, z_2, \ldots\ldots z_m]$ as the input and output variables.

$$\hat{z} = f(W) = \frac{1}{N}\sum_{n=1}^{N} f_n(W) \tag{10}$$

$f_n(W)$ and N represents the prediction made by the n^{th} model and the total number of base learners in the stacked ensemble.

3.3.2. Enhancing inventory optimization in supply chains with SE-RGLR

Inventory management or inventory planning [31] is one of the major issues in SCM, it is basically focused on achieving an optimal level of production and inventory to ensure that holding costs are minimized while the costs associated with stock-out conditions are also taken into consideration. The SE-RGLR agent constantly observes the state of the supply chain analyzing quantities of inventories, trends in demand, and performance of suppliers. By interacting with the environment, the agent realizes a need to maintain efficient stock quantities, keeping costs pertaining to unnecessary stock high while also is not deprived of the impact of stockouts such as sales that are lost or backorders. The model employs a policy that can translate the current supply chain situation to the appropriate course of action to optimize inventory. Inventory management is a function designed to make several efforts to reduce costs while still meeting necessary stock levels. The objective of the inventory optimization component is to model total inventory-related costs D which can be defined as:

$$D = G.J_s + T.max(0, C_s - J_s) \tag{11}$$

where J_s, $\hat{z}$,G, C_s and T signifies optimal inventory levels at a time s, demand forecasts, holding costs, predicted demand, and stockout costs.

3.3.3. Improving procurement decisions through supplier performance prediction with SE-RGLR

Measuring the performance of a supplier [7] is an important aspect of the supply chain as it affects the quality of a product to be offered in the market, the delivery time, and the efficiency of operations. The SE-RGLR model enables efficient supplier performance assessments by using data from key past supplier performance indicators including lead time, defect rate, and delivery reliability. The SE-RGLR agent uses the data generated by previous interactions with suppliers to adjust for current and future behavior. It targets suppliers that may be on time, deliver quality products or services or are less likely to be late. To do this, the method integrates changes in supply chain predictions with new data in their environment. To predict supplier efficacy from preceding records the developed SE-RGLR approach is utilized. The technique learns the probability $O(t)$ that a supplier fulfils an order on time with acceptable quality given past effective measures:

$$O(t) = \sigma_a\left(\sum_{j=1}^{l} x_j w_j\right) \tag{12}$$

w_j, x_j, and σ_a represents supplier performance, the allocated weight to every feature, and the sigmoid function, ensuring the output $O(t)$ lies between 0 and 1.

3.3.4. Streamlining supply chain operations for logistics optimization with SE-RGLR

In the supply chain, logistics management is a momentous factor in handling various tasks including storage facilities management, transportation vehicles, and transport planning. These activities need cost control and effective delivery strategies that aim to improve the overall organization's work. The SE-RGLR model integrates uncertainty into decision-making processes to improve the logistical concerns such as route selection, vehicle allocation mechanisms, and choice of transportation modes and types all within the context

of cost, time, inventory, and capacity constraints. The model, thus, redetermines strategies that improve the model of logistical efficiency through habitual interaction with the supply chain environment. The SE-RGLR model involves feedback loops to refine decision-making at successive steps of the process. This is because it can always assess results and change strategies fulfilling needs that may change periodically or responding to shocks whenever they arise. Key logistics decisions include:

a. Route planning: the SE-RGLR model goes in for the best delivery routes by involving parameters such as distance, congestion in the road, and the dragging capacity of the vehicle. It minimizes costs related to transport while promoting timely delivery thus improving the overall organization of delivery.
b. Fleet management: the SE-RGLR model improves management by estimating the optimum number and variety of vehicles required to accomplish all delivery tasks. It also helps to reduce costs considerably and maximize the benefits derived from available resources without further delay.
c. Warehouse operations: by moving goods into better organizational zones, minimizing travel distances, and making order fulfilment more efficient, the SE-RGLR model altogether improves warehouse management. These strategies are intended to support efficient warehouse organization and operational planning.
d. Transportation scheduling: the SE-RGLR model also improves the schedule of transporting by recognizing the ideal time to transport, integrating modes of transport, and tackling transportation resources for differing hubs. This results in better delivery time and cost rates.

Through coordinated modelling of logistics functions, the SE-RGLR framework supports structured decision-making under cost, time, and capacity constraints. It helps organizations to maintain adequate competitiveness and supply chain responsiveness to alterations of the existing logistics restrictions and other market conditions. The step-by-step implementation of this process, including data preprocessing, base model training, ensemble construction, and prediction generation, is summarized in Algorithm 1.

Algorithm 1. Pseudocode for the developed SE-RGLR model

```
Input: Supply-chain dataset (D)
Output: Final SE-RGLR predictions (Ŷ)
1. Start
2. Load the supply-chain dataset (D).
3. Identify the input features (X) and target variable (Y).
4. Clean the dataset by:
    handling missing values using median imputation;
    removing duplicate observations;
    encoding categorical variables;
    transforming and normalizing numerical features;
    identifying and handling outliers.
5. Partition the preprocessed dataset into training and testing subsets using the fixed
   80:20 split.
6. Initialize the base learners:
    Random Forest (RF);
    Gradient Boosted Decision Trees (GBDT);
    Logistic Regression (LR).
7.  Divide the training set into (K) folds for generation of out-of-fold predictions.
8.  For (k=1, … ,K):
     a. Train RF, GBDT, and LR using the other (K-1\) folds.
     b. Generate predictions for the held-out (k^th) fold.
     c. Store the predictions as P_RF^(k), P_GBDT^k, and (P_LR^k).
9.  Combine the predictions from all folds to construct the OOF meta-feature matrix:
        Z = [P_RF^(k), P_GBDT^k, (P_LR^k)]
10. Train the Logistic Regression meta-learner using (Z) and the corresponding training
    labels (Y).
11. Estimate the ensemble coefficients β0, β1, β2, β3 by minimizing the regularized logistic
    loss.
12.  Retrain RF, GBDT, and LR using the complete training dataset.
13.  Generate predictions of the three retrained base learners on the independent test
    dataset.
14. Construct the test meta-feature matrix:
        Z_test=[P_RF,test, P_GBDT,test, P_LR,test]
15.  Pass (Z_test) to the trained meta-learner.
16.  Generate the final SE-RGLR predictions (Ŷ_test).
17.  Evaluate the final predictions using the predefined performance metrics.
18.  Compare SE-RGLR performance with the individual base learners.
19.  End
```

Model parameters were kept consistent across experiments to isolate the impact of the proposed ensemble strategy on forecasting performance.

4. EXPERIMENTAL RESULTS

This section explains a detailed evaluation of the developed method; this section begins with outlining the experimental configuration, parameter settings, and evaluation metrics are presented, which are used to analyze the model's efficacy. Moreover, the effectiveness of the developed method is calculated using certain metrics and compared with existing approaches.

To ensure a fair and consistent evaluation, all experiments were conducted using a fixed train–test evaluation protocol. The dataset was split into training and testing subsets in an 80:20 ratio, as described in the methodology section. All comparative models, including the proposed SE-RGLR framework, were trained on the same training data and evaluated on the same test data. No cross-validation or repeated runs were employed, and all performance metrics are reported consistently across models to enable direct comparison.

4.1. Experimental configuration

The experiments are conducted on a personal computer with the specifications of an Intel Core i7 processor, 16 GB of RAM, and a 512 GB SSD, providing computational power and storage for efficient data processing. The software environment consisted of Python 3.9 executed within Jupyter Notebook, with standard machine learning libraries. All experiments were conducted on a personal computer equipped with an Intel Core i7 processor, 32 GB RAM, and a 1 TB SSD. To ensure reproducibility, a fixed random seed was used across all experiments. Table 1 lists the optimal values of every parameter.

Table 1. Listed the optimal values of every parameter

Parameters	Values
Number of trees ($n_estimators$)	100
Maximum depth (max_depth)	10
Learning rate ($learning_rate$)	0.1
Regularization (C)	1.0
Penalty	L2 (ridge regularization)

4.2. Evaluation metrics

To estimate the effectiveness of the SE-RGLR model across various techniques the evaluation measures such as accuracy, precision, recall, specificity, F1-score, area under the receiver operating characteristic curve (AUC-ROC), mean absolute error (MAE), root mean squared error (RMSE), and MSE are used. The mathematical expressions and definitions are illustrated as:

It computed the count of correct predictions and is mathematically represented using (13).

$$Accuracy = \frac{Tr_{Pe}+Tr_{Ne}}{Tr_{Pe}+Tr_{Ne}+Fs_{Pe}+Fs_{Ne}} \quad (13)$$

The ratio between the total count of negative samples to the count of samples tested as negative in the experiment. Specificity is shown in (14).

$$S_p = \frac{Tr_{Ne}}{Tr_{Ps}+Tr_{Ne}} \quad (14)$$

Precision represents the percentage of true positives over the entire samples predicted as positive. The computation of precision is done using (15).

$$P_{re} = \frac{Tr_{Ps}}{Tr_{Ps}+Fs_{Ps}} \quad (15)$$

This refers to the proportion of actual positive cases that are correctly identified by the model.

$$R_c = \frac{Tr_{Ps}}{Tr_{Ps}+Fs_{Ne}} \quad (16)$$

The measurement of the harmonic mean of precision and recall is known as the F1-score. The computation of the F1 score is done using (17).

$$F1_s = 2 \times \frac{P_{re} \times R_c}{P_{re}+R_c} \quad (17)$$

AUC-ROC: the trade-off between the true and false positives is indicated by the AUC-ROC. It is evaluated using the following (18)-(20). MAE is calculated by the mean variance among the predicted value and important worth. In (18) provides the mathematical presentation of MAE:

$$MAE = \frac{1}{m}\sum_{j=1}^{m}\left|z_j - \hat{z}_j\right| \quad (18)$$

z_j and $\hat{z}_j$ is the true value and predicted value.

MSE is calculated by the mean of square values of the actual and estimated values' difference. It measures the error amount in numerical ways.

$$MSE = \frac{1}{m}\sum_{j=1}^{m}\left(z_j - \hat{z}_j\right)^2 \quad (19)$$

RMSE is calculated by radicand of square values of the entire errors' mean. Each error of RMSE's impact is proportional to the size of square values of the errors.

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(z_j - \hat{z}_j\right)^2} \quad (20)$$

4.3. Experimental outcome

The performance analysis is a performance for displaying the efficiency of the proposed model by using different performance measures and comprehensive analyses. The performance attained by the proposed technique in terms of all the evaluation metrics used is presented in Table 2.

Table 2. Performance assessment on different metrics

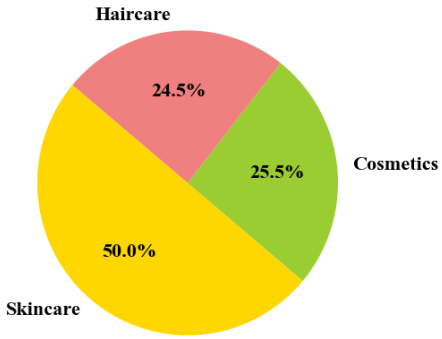
Parameters	Values
Accuracy	98.6
Precision	97.5
Recall	99.4
Specificity	98.5
F1-score	99.2
RMSE	0.82
MAE	0.78
MSE	0.79

The strong performance of the proposed SE-RGLR model across all evaluation metrics can be attributed to the complementary learning behavior of its base learners. RF effectively captures non-linear feature interactions and reduces variance through bagging, while GBDT enhance predictive accuracy by sequentially correcting residual errors. The LR meta-learner integrates these diverse predictions into a unified decision boundary, leading to improved generalization and robustness compared to individual models.

Analyzing sales by product type acquires valuable insights into how revenue is distributed across a company's portfolio and helps assess the performance of different product categories. Figure 3 illustrates the sales distribution across product categories: skincare products lead with 50% of total sales revenue, followed by hair care products at 25.5%, and cosmetics at 24.5%. This distribution emphasizes skincare as the most profitable category, suggesting strong consumer demand and potentially higher profit margins compared to other products.

Figure 4 displays the income generated by every shipping carrier, providing a clear graphical representation that facilitates quick identification of the top-performing carriers. Assessing the revenue from shipping carriers is crucial for effective carrier selection and management in supply chain operations. X-axis represents shipping carriers (A, B, C) and Y-axis shows the revenue generated by each carrier in units. Carrier B generates the highest revenue of 250 units, and carriers A and C generate comparatively low revenues of 151 and 170 units.

Figure 5 illustrates the expenses and defect ratio corresponding to different modes of transportation: road, air, and rail. X-axis represents transportation modes (road, air, and rail) and Y-axis shows the corresponding cost incurred in units. Road transportation is the most cost-effective, but its cost varies across scenarios based on distance and route conditions which attained 120.2 and 230.5 units. Rail transportation incurs a higher cost than road but is less expensive than air transport and achieved 430.3 units this suggests a possible correlation between cost and defect rate. Air transportation has the highest cost variability which attained 320.5 and 550.5 units reflecting the influence of factors like fuel, distance, and premium services.



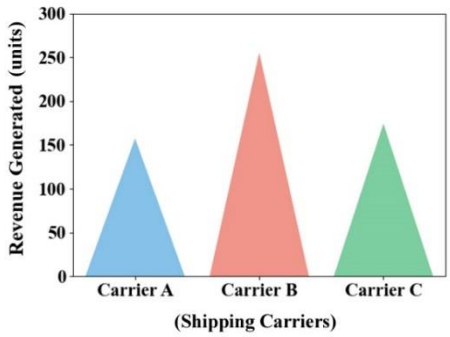


Figure 3. Product type sales distribution analysis

Figure 4. Revenue analysis generated by shipping careers

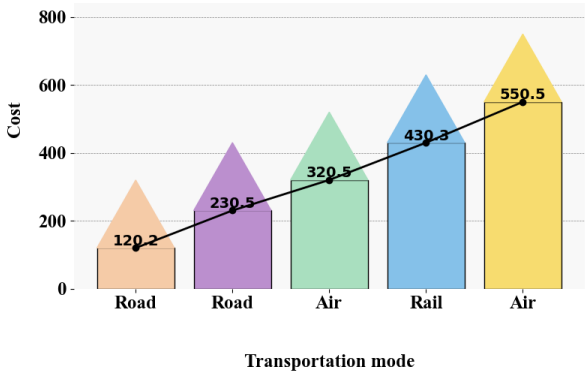


Figure 5. Cost distribution analysis

Figure 6 presents two plots: Figure 6(a) the AUC-ROC curve and Figure 6(b) the precision-recall (P-R) curve, both of which are used to assess the performance of the proposed machine-learning model. Figure 6(a) presents the ROC curve with an AUC of 0.97, indicating strong discrimination between the evaluated classes. The P-R curve indicates high precision at low recall levels, suggesting that the classifier is effective at identifying positive instances with high confidence. However, as recall increases, the precision decreases, implying that the classifier may miss some positive instances as it becomes more inclusive in its predictions. The confusion matrix of SE-RGLR model is shown in Figure 7, which shows the distribution of correctly and incorrectly classified instances for the evaluated classes. The matrix gives us more than just the overall performance of how the model classifies.

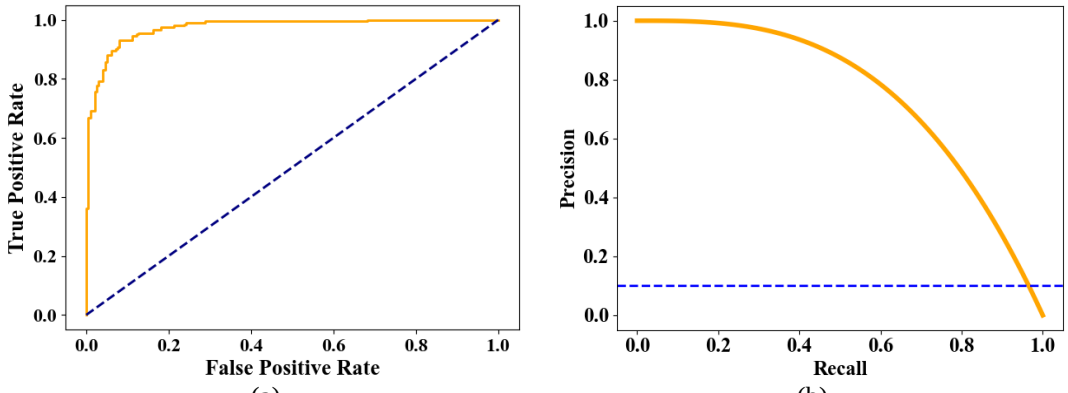


Figure 6. Analysis of (a) AUC-ROC and (b) P-R curve

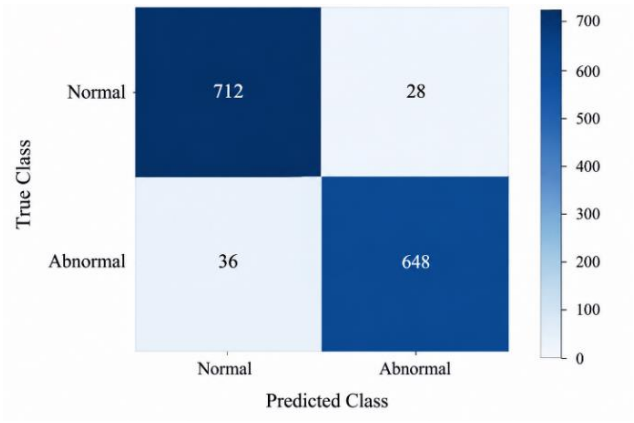


Figure 7. Confusion matrix of SE-RGLR model

4.4. Comparative results

The comparative analysis of the proposed technique SE-RGLR with the existing techniques such as UDAF [15], PCGS [21], PPSCN [14], JLSTM [17], and SGRU [19] is presented in detail with neat graphical representations in this section. The comparative analysis of the proposed method as well as existing methods on diverse evaluation measures is displayed in Figure 8. When compared with UDAF, PCGS, PPSCN, JLSTM, and SGRU, the developed method attained the highest accuracy, precision, recall, specificity, and F1-score of 98.6%, 97.5%, 99.4%, 98.5%, and 99.2%. Overall, the below graphical representation displays the proposed SE-RGLR framework obtained excellent outcomes in all evaluation measures. This indicates the efficiency of the proposed strategy in SCM. The selected comparison methods represent different methodological categories, including machine-learning-based, deep-learning-based, and hybrid optimization approaches. This comparison provides a broader assessment of SE-RGLR across diverse forecasting and supply chain modelling strategies.

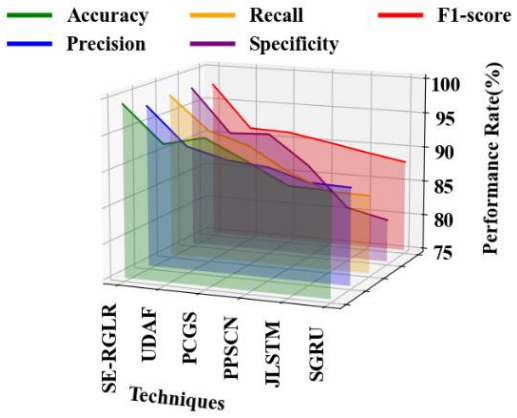


Figure 8. Comparative analysis of accuracy, precision, recall, specificity, and F1-score

The RMSE, MAE, and MSE analysis of the proposed and existing methods are illustrated in Figure 9. The proposed strategy attained the lowest values of 0.82, 0.78, and 0.79 in the context of RMSE, MAE, as well as MSE. This analysis provided that the proposed SE-RGLR method attained a better performance in SCM when compared to other existing methodologies.

The results are also relevant to intelligent engineering systems. In IoT-enabled SCM environments, continuous data on inventory and logistics can be transferred via reliable communication networks and processed by the forecasting framework to be able to make timely decisions. From a control-system perspective, more accurate demand forecasts could lead to more stable inputs for inventory and logistics control decisions, and cloud and edge computing could enable scalable deployment of the proposed analytics framework.

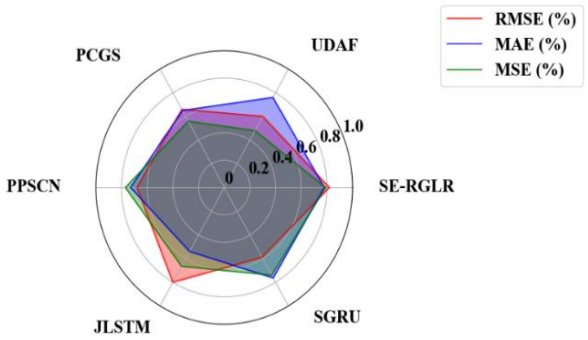


Figure 9. Comparative analysis of RMSE, MAE, and MSE

The superior performance of SE-RGLR over existing methods arises from its ensemble diversity and stacked learning strategy. Unlike single-model approaches, the proposed framework combines multiple learners with distinct inductive biases, enabling it to handle heterogeneous supply chain data more effectively. This ensemble structure reduces overfitting and prediction variance while leveraging complementary strengths, which explains the consistent improvements observed across classification and error-based metrics in comparison with UDAF, PCGS, PPSCN, JLSTM, and SGRU. Table 3 gives the comparative analysis of SE-RGLR with existing approaches.

Table 3. Comparative analysis of SE-RGLR with UDAF, PCGS, PPSCN, JLSTM, and SGRU

Model	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)	RMSE	MAE	MSE
SE-RGLR	98.6	97.5	99.4	98.5	99.2	0.82	0.78	0.79
UDAF	92.4	90.8	93.1	94.7	91.9	0.91	0.86	0.84
PCGS	91.7	89.9	94.2	93.8	91.9	0.94	0.88	0.87
PPSCN	88.9	87.2	89.7	91.5	88.4	1.02	0.96	0.93
JLSTM	86.5	85.8	87.4	88.6	86.6	1.08	1.01	0.97
SGRU	84.8	84.6	85.9	82.7	85.1	1.13	1.05	1.01

4.5. Ablation study

An ablation study for the SE-RGLR model, as shown in Table 4, assesses the impact of various components and features on its performance by systematically removing or altering them. The study compares the full model, which combines RF, GBDT, and LRT in a stacked ensemble, with versions where individual components are omitted. The consequences listed that the full model delivers the superior performance, with a slight decline in accuracy, sensitivity, and MAE when either RF or GBDT is removed. The removal of the stacked ensemble mechanism results in the most significant performance drop, underscoring the importance of combining multiple models. The performance gain demonstrates that model diversity and ensemble aggregation significantly reduce prediction variance and improve generalization. Feature ablation, such as removing supplier or transportation data, leads to a moderate performance decrease, suggesting that these features are valuable but not essential.

Table 4. Ablation analysis

Models	Accuracy	Precision	Recall	Specificity	F1-score	RMSE	MSE	MAE	Stockout rate (%)
SE-RGLR	98.6	97.5	99.4	98.5	99.2	0.82	0.79	0.78	62
Remove RF	88.6	85.2	84.7	90.7	84.9	0.83	0.80	0.81	54.5
Remove GBDT	89.1	86.5	85.4	91.8	85.4	0.85	0.82	0.79	55
Remove LRT	87.5	84.4	83.8	89.5	83.5	0.88	0.87	0.83	54
Remove stacked ensemble	85.4	87.4	82.2	88.3	82.3	0.84	0.85	0.82	53.5
Remove supplier features	90.9	87.2	86.5	92.6	86.2	0.87	0.86	0.87	55.7
Remove transportation features	91.6	88.1	87.6	93.2	87.6	0.85	0.90	0.84	56.3

Component-level ablation results show that removing individual base learners leads to a decrease in the overall predictive performance, while removal of the stacked ensemble leads to a large decrease in performance. These observations show the complementary contribution of the constituent learners and the ensemble aggregation mechanism.

From a practical supply chain perspective, the results suggest that enhanced demand forecasting can facilitate better inventory planning and reduce the risk of overstocking and stockouts. Supplier related predictions can help organizations to identify potential delivery and quality issues. Logistics related analysis can help with transportation and routing decisions. These applications provide a foundation for improved inventory utilization, supplier reliability evaluation and logistics efficiency.

4.6. Statistical results

The Wilcoxon signed-rank test was conducted to statistically validate the performance differences observed between the proposed SE-RGLR model and existing methods. This non-parametric test evaluates whether the improvements reported in the comparative performance tables are statistically significant rather than arising from random variation. The results of the Wilcoxon signed-rank test are presented in Table 5, which reports pairwise comparisons and corresponding p-values. A p-value less than 0.05 indicates statistically significant performance differences, confirming the reliability of the improvements achieved by the proposed approach.

Table 5. Wilcoxon signed-rank test (two-tailed)

Pairwise model comparison		p-value	Significance < 0.5
Model 1	Model 2		
SE-RGLR	UDAF	0.0001	Yes
SE-RGLR	PCGS	0.0117	Yes
SE-RGLR	PPSCN	0.0001	Yes
SE-RGLR	JLSTM	0.0001	Yes
SE-RGLR	SGRU	0.0030	Yes
UDAF	PCGS	0.0001	Yes
UDAF	PPSCN	0.0028	Yes
UDAF	JLSTM	0.0001	Yes
UDAF	SGRU	0.0001	Yes
PCGS	PPSCN	0.0068	Yes
PCGS	JLSTM	0.0023	Yes
PCGS	SGRU	0.0052	Yes
PPSCN	JLSTM	0.0001	Yes
PPSCN	SGRU	0.0045	Yes
JLSTM	SGRU	0.0001	Yes

5. CONCLUSION

This study demonstrates the effectiveness of a stacked ensemble learning framework for demand forecasting and operational optimization in SCM within the fashion and beauty industry. By integrating multiple machine learning models, the proposed SE-RGLR approach provides accurate and reliable forecasts that support improved inventory planning, supplier evaluation, and logistics decision-making. Experimental results indicate that the proposed model achieves strong predictive performance, attaining an accuracy of 98.6%, precision of 97.5%, recall of 99.4%, specificity of 98.5%, and an F1-score of 99.2%, while maintaining low error values (RMSE = 0.82, MAE = 0.78, and MSE = 0.79). These improvements translate into tangible managerial benefits, including reduced stockouts, lower operational costs, and enhanced responsiveness to demand variability. The proposed SE-RGLR model also demonstrates strong compatibility with IoT-based sensing platforms, cloud and edge computing architectures, and intelligent control systems highlighting its relevance to modern electrical and telecommunication infrastructures. Despite these promising results, the study is limited to a single industry-specific dataset and does not consider real-time deployment constraints or interpretability aspects. Future work will focus on extending the framework to multi-industry supply chains, incorporating real-time data streams from IoT-enabled systems, and exploring advanced deep learning models to further improve forecasting robustness and scalability

ACKNOWLEDGMENTS

The project was supported by KAU Endowment (WAQF) at king Abdulaziz University, Jeddah, Saudi Arabia. The authors therefore acknowledge with thanks WAQF and the Deanship of Scientific Research (DSR) for technical and financial support.

FUNDING INFORMATION

Authors state that no funding was received for this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Norah Nasser M ALQutaim	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Abdulrhman ALShareef		✓				✓		✓	✓		✓			
Syed Hamid Hasan	✓		✓	✓			✓			✓	✓		✓	
Usman Ali Khan			✓		✓	✓			✓			✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/amirmotefaker/supply-chain-dataset.

REFERENCES

[1] Y. Khan, M. B. M. Su'ud, M. M. Alam, S. F. Ahmad, A. Y. A. B. Ahmad, and N. Khan, "Application of Internet of Things (IoT) in Sustainable Supply Chain Management," *Sustainability (Switzerland)*, vol. 15, no. 1, p. 694, 2023, doi: 10.3390/su15010694.

[2] N. J. Javid and M. Amini, "Evaluating the effect of supply chain management practice on implementation of halal agroindustry and competitive advantage for small and medium enterprises," *International Journal of Computer Science and Information Technology*, vol. 15, no. 6, pp. 8997–9008, 2023.

[3] X. M. Yuan and A. Xue, "Supply Chain 4.0: New Generation of Supply Chain Management," *Logistics*, vol. 7, no. 1, p. 9, 2023, doi: 10.3390/logistics7010009.

[4] S. Shrivastava, "Recent trends in supply chain management of business-to-business firms: a review and future research directions," *Journal of Business and Industrial Marketing*, vol. 38, no. 12, pp. 2673–2693, 2023, doi: 10.1108/JBIM-02-2023-0122.

[5] O. M. Al-Rawashdeh, O. Jawabreh, and B. J. A. Ali, "Supply Chain Management and Organizational Performance: The Moderating Effect of Supply Chain Complexity," *Information Sciences Letters*, vol. 12, no. 3, pp. 1673–1684, 2023.

[6] S. Ferraro, L. Leoni, A. Cantini, and F. De Carlo, "Trends and Recommendations for Enhancing Maturity Models in Supply Chain Management and Logistics," *Applied Sciences (Switzerland)*, vol. 13, no. 17, p. 9724, 2023, doi: 10.3390/app13179724.

[7] N. Letunovska, F. A. Offei, P. A. Junior, O. Lyulyov, T. Pimonenko, and A. Kwilinski, "Green Supply Chain Management: The Effect of Procurement Sustainability on Reverse Logistics," *Logistics*, vol. 7, no. 3, p. 47, 2023, doi: 10.3390/logistics7030047.

[8] A. Attia, "Effect of Sustainable Supply Chain Management and Customer Relationship Management on Organizational Performance in the Context of the Egyptian Textile Industry," *Sustainability (Switzerland)*, vol. 15, no. 5, p. 4072, 2023, doi: 10.3390/su15054072.

[9] J. Zaman, A. Shoomal, M. Jahanbakht, and D. Ozay, "Driving Supply Chain Transformation with IoT and AI Integration: A Dual Approach Using Bibliometric Analysis and Topic Modeling," *Internet of Things*, vol. 6, no. 2, p. 21, 2025, doi: 10.3390/iot6020021.

[10] A. A. Alshdadi, A. A. Almazroi, and N. Ayub, "IoT-driven load forecasting with machine learning for logistics planning," *Internet of Things (The Netherlands)*, vol. 29, p. 101441, 2025, doi: 10.1016/j.iot.2024.101441.

[11] K. C. Yarlagadda, "AI-Powered Supply Chain Optimization: Enhancing Demand Forecasting and Logistics," *Journal of Computer Science and Technology Studies*, vol. 7, no. 4, pp. 792–801, 2025, doi: 10.32996/jcsts.2025.7.4.92.

[12] M. Shokouhifar, M. Sohrabi, M. Rabbani, S. M. H. Molana, and F. Werner, "Sustainable Phosphorus Fertilizer Supply Chain Management to Improve Crop Yield and P Use Efficiency Using an Ensemble Heuristic–Metaheuristic Optimization Algorithm," *Agronomy*, vol. 13, no. 2, p. 565, 2023, doi: 10.3390/agronomy13020565.

[13] Z. Benmamoun, K. Khlie, M. Dehghani, and Y. Gherabi, "WOA: Wombat Optimization Algorithm for Solving Supply Chain Optimization Problems," *Mathematics*, vol. 12, no. 7, p. 1059, 2024, doi: 10.3390/math12071059.

[14] L. Pan, M. Shan, and L. Li, "Optimizing Perishable Product Supply Chain Network Using Hybrid Metaheuristic Algorithms," *Sustainability (Switzerland)*, vol. 15, no. 13, p. 10711, 2023, doi: 10.3390/su151310711.

[15] E. Aldahmani, A. Alzubi, and K. Iyiola, "Demand Forecasting in Supply Chain Using Uni-Regression Deep Approximate Forecasting Model," *Applied Sciences (Switzerland)*, vol. 14, no. 18, p. 8110, 2024, doi: 10.3390/app14188110.

[16] C. C. Wang, H. T. Chang, and C. H. Chien, "Hybrid LSTM-ARMA Demand-Forecasting Model Based on Error Compensation for Integrated Circuit Tray Manufacturing," *Mathematics*, vol. 10, no. 13, p. 2158, 2022, doi: 10.3390/math10132158.

[17] R. Khalid, N. Javaid, F. A. Al-zahrani, K. Aurangzeb, E. U. H. Qazi, and T. Ashfaq, "Electricity load and price forecasting using

jaya-long short term memory (JLSTM) in smart grids," *Entropy*, vol. 22, no. 1, p. 10, 2020, doi: 10.3390/e22010010.
[18] M. M. Eid *et al.*, "Meta-Heuristic Optimization of LSTM-Based Deep Network for Boosting the Prediction of Monkeypox Cases," *Mathematics*, vol. 10, no. 20, p. 3845, 2022, doi: 10.3390/math10203845.
[19] S. M. Abdullah *et al.*, "Optimizing Traffic Flow in Smart Cities: Soft GRU-Based Recurrent Neural Networks for Enhanced Congestion Prediction Using Deep Learning," *Sustainability (Switzerland)*, vol. 15, no. 7, p. 5949, 2023, doi: 10.3390/su15075949.
[20] T. S. C. Chetty *et al.*, "Optimized Hierarchical Tree Deep Convolutional Neural Network of a Tree-Based Workload Prediction Scheme for Enhancing Power Efficiency in Cloud Computing," *Energies*, vol. 16, no. 6, p. 2900, 2023, doi: 10.3390/en16062900.
[21] B. Aljabhan and M. A. Obaidat, "Privacy-Preserving Blockchain Framework for Supply Chain Management: Perceptive Craving Game Search Optimization (PCGSO)," *Sustainability (Switzerland)*, vol. 15, no. 8, p. 6905, 2023, doi: 10.3390/su15086905.
[22] S. Shukla and S. Pasari, "Short-term wind speed prediction with adaptive signal processing based hybrid statistical models," *Energy Systems*, vol. 17, no. 4, pp. 1691–1721, 2026, doi: 10.1007/s12667-025-00727-6.
[23] X. Hu, "Cold chain logistics model of agricultural products based on embedded system and blockchain," *Production Planning & Control*, pp. 1–12, 2022, doi: 10.1080/09537287.2022.2101939.
[24] B. Amangeldy, N. Tasmurzayev, S. Shinassylov, A. Mukhanbet, and Y. Nurakhov, "Integrating Machine Learning with Intelligent Control Systems for Flow Rate Forecasting in Oil Well Operations," *Automation*, vol. 5, no. 3, pp. 343–359, 2024, doi: 10.3390/automation5030021.
[25] A. Motefaker, "Supply chain dataset," *Kaggle Contributor*, 2023. Accessed: Feb. 10, 2026. [Online]. Available: https://www.kaggle.com/datasets/amirmotefaker/supply-chain-dataset
[26] S. Das, M. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, "A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier," *Expert Systems with Applications*, vol. 213, p. 118914, 2023, doi: 10.1016/j.eswa.2022.118914.
[27] W. Cai, R. Wei, L. Xu, and X. Ding, "A method for modelling greenhouse temperature using gradient boost decision tree," *Information Processing in Agriculture*, vol. 9, no. 3, pp. 343–354, 2022, doi: 10.1016/j.inpa.2021.08.004.
[28] H. Shang *et al.*, "Spatial Prediction of Landslide Susceptibility Using Logistic Regression (LR), Functional Trees (FTs), and Random Subspace Functional Trees (RSFTs) for Pengyang County, China," *Remote Sensing*, vol. 15, no. 20, p. 4952, 2023, doi: 10.3390/rs15204952.
[29] R. Kablan, H. A. Miller, S. Suliman, and H. B. Frieboes, "Evaluation of stacked ensemble model performance to predict clinical outcomes: A COVID-19 study," *International Journal of Medical Informatics*, vol. 175, p. 105090, 2023, doi: 10.1016/j.ijmedinf.2023.105090.
[30] A. R. Muthukalyani, "Unlocking Accurate Demand Forecasting in Retail Supply Chains with AI-driven Predictive Analytics," *International Journal of Information Technology and Management Information Systems (IJITMIS)*, vol. 14, no. 2, pp. 48–57, 2023.
[31] M. Seyedan, F. Mafakheri, and C. Wang, "Order-up-to-level inventory optimization model using time-series demand forecasting with ensemble deep learning," *Supply Chain Analytics*, vol. 3, p. 100024, 2023, doi: 10.1016/j.sca.2023.100024.

BIOGRAPHIES OF AUTHORS



Norah Nasser M ALQutaim obtained her Bachelor's degree in Computer Science from King Faisal University Then her Master's degree in Business Information System Management from Middlesex University, London, United Kingdom. She is currently a lecturer at the Faculty of Information system Management, Hail University, her research interests are machine learning, relationship between information technology and business. She can be contacted at email: nalqutaim@stu.kau.edu.sa.



Abdulrhman Alshareef is an Associate Professor in the Information Systems Department at King Abdulaziz University. He holds a Ph.D. in Computer Science from the University of Ottawa and specializes in data science, machine learning, and secure software development. He has led numerous AI-focused research projects and plays a key role in curriculum development, accreditation, and academic quality enhancement. His work connects advanced research with practical applications in AI and information systems. He can be contacted at email: amralshareef@kau.edu.sa.



Syed Hamid Hasan is a senior academic at King Abdulaziz University, Jeddah, with extensive experience in computer science and information systems. His research interests include cybersecurity, cloud computing, artificial intelligence, and data science. He has taught and mentored students at institutions such as Aligarh Muslim University and the College of Technology, Oman. He has authored over 60 research publications and led projects on secure smart grids, AI-based energy forecasting, and cloud optimization. An IEEE member, he is recognized for integrating academic rigor with practical technological applications. He can be contacted at email: shhasan@kau.edu.sa.



Usman Ali Khan is an Associate Professor in the Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. He received his Ph.D. in Software Engineering from the Integral University, India. He has been working is the field of research and teaching at graduate and undergraduate level in different Universities since 1995. His current research interest includes AI, machine learning, software engineering and project management He has published many research papers at national and international forums. He can be contacted at email: ukhan@kau.edu.sa.