Journal of Industrial and Management Optimization

JIMO, 22 (1): 374–404.
DOI: 10.3934/jimo.2026014
Received: 28 April 2025
Revised: 13 August 2025
Accepted: 18 September 2025
Published: 03 December 2025

https://www.aimspress.com/journal/jimo

Research article

A fractal deep learning model for optimizing e-commerce warehousing and logistics

Huomei Zhou*, Wenyu Ning and Tao Guo

School of Economics and Management, Jiujiang Polytechnic University of Science and Technology, Gongqing 332020, China

* **Correspondence:** Email: huomei_zhou@outlook.com.

Abstract: E-commerce warehousing and logistics optimization are crucial in meeting increased consumer demand, reducing costs, and improving operational efficiency. Previous work has applied machine learning and deep learning techniques to improve inventory management, demand forecasting, and route optimization in logistics. However, the challenges of scalability and adaptability remain, mainly in dealing with complicated and dynamic e-commerce warehousing operations. This research proposes the Fractal Deep Learning Model with the Particle Swarm Optimization algorithm (FDLM-PSO) to upgrade the efficacy and adaptability in e-commerce warehousing and logistics. In the FDLM, fractal neural networks support hierarchical learning and processing of multidimensional data related to inventory levels, real-time delivery schedules, and geographic constraints. PSO is a strong optimization technique for dynamic route planning under variable constraints, like traffic and customer priorities, ensuring efficient delivery schedules. Additionally, reinforcement learning is integrated with adaptive inventory control, enabling the real-time creation of decisions aiming at stockouts or overstocking reduction. Results from the proposed model show a 20% reduction in inventory holding cost, a 25% enhancement in delivery time prediction, and a 20% improvement in route efficiency compared to conventional methods. The paper's ability to scale and adapt to high-demand fluctuations attests to its robustness and applicability. The proposed FDLM-PSO addresses critical gaps in the current methodology, developing a scalable and efficient framework for optimizing e-commerce logistics and warehousing operations.

Keywords: fractal deep learning model; e-commerce logistics optimization; particle swarm optimization; adaptive inventory control; dynamic route planning; warehousing efficiency
Mathematics Subject Classification: 68T07, 68U35

1. Introduction

E-commerce has undergone a renaissance in the last couple of years, ingraining itself in world trade and consumer lifestyles. This has brought about significant transformation in technology, logistics, and supply chain management to better cater to the ever-growing demands of an online consumer [1]. With consumers expecting shorter delivery times, personalization of services, and seamless experiences, there is an increasing challenge for e-commerce companies in optimizing warehousing and logistics operations. These challenges include increased supply chain and logistics network complexity, which calls for innovative solutions to ensure efficiency, scalability, and adaptability [2].

1.1. Evolution of e-commerce logistics and warehousing

Supply chain logistics management and warehousing are shown in Figure 1. Manufacturers, distributors, wholesalers, and retailers are the main intermediates in the supply chain from raw material suppliers to end consumers. Receiving, storing, sorting, order selecting, and shipping are covered in the lower half. Both methods ensure inventory handling efficiency, storage optimization, and on-time supply chain delivery, as shown in the Figure 1.

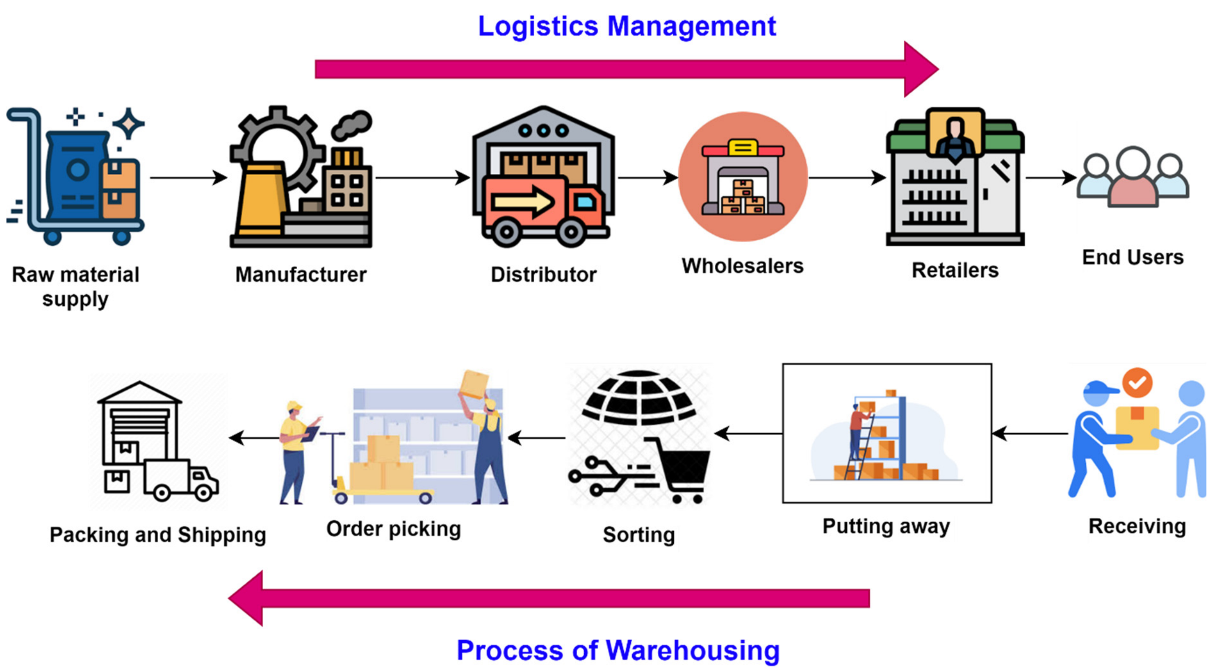


Figure 1. E-commerce logistics management and warehouse process.

1.2. Challenges in traditional logistics models

E-commerce has disrupted logistics and warehousing methods due to its rapid expansion. Warehouses used manual methods and static storage policies to handle predictable and steady inventory flows [3]. Modern e-commerce's high quantities and speeds are too much for such setups. With rising customer expectations for fast and seamless order delivery, scaling existing approaches is no longer practicable, resulting in inefficiencies, bottlenecks, and higher prices [4]. Manual functioning cannot adapt to real-time demand changes or supply chain problems. Inefficiencies are exacerbated by static storage strategies, which place products regardless of demand [5]. Traditional methods don't work for e-commerce, thus dynamic and data-driven logistics and warehousing are needed [6].

1.3. Technological advancements and modern paradigms

New paradigm-shifting logistics and warehousing technologies overcome these limits. Warehouse automation, robotics, and conveyor systems have enhanced efficiency and accuracy [6]. IoT devices and powerful software allow organizations to track inventory levels in real time and make data-driven choices. Advanced algorithms and geographical data optimise delivery timetables by reacting to real-time variables like traffic and priorities [7].

Amazon, Alibaba, and Walmart have developed new logistics models using these technologies. Data-driven decision-making and automation have improved operational efficiency and customer service at these organizations [8]. Amazon's fulfillment centers use robotics for picking and packing, while Alibaba's Cainiao network leverages IoT and big data analytics for supply chain management. Robotics can make logistics and warehousing more flexible and efficient [9].

1.4. Artificial intelligence and machine learning in logistics

Recently, artificial intelligence and machine learning have revolutionized logistics and warehousing. Companies may handle massive amounts of data from many sources using AI to gain relevant insights and automate decision-making [10]. Considerations including past trends, customer behavior, and seasonality have helped machine learning models predict demand. Businesses can optimize storage allocation and eliminate operational inefficiencies with ML demand predictions [11]. Predictive analytics can detect popular products and ensure availability at crucial warehouse sites to avoid stockouts and delivery delays. Dynamic space allocation based on product turnover and demand variability improves storage efficiency with AI-driven optimization methods [12].

Due to restricted AI and ML technology, most e-commerce enterprises cannot scale. Traditional AI models struggle with multidimensional data from inventory levels to delivery schedules with geography and consumer preference limitations [13]. AI and ML solutions require a lot of computer power, making implementation in SMEs problematic. These issues require innovative systems that are scalable, adaptable, and can analyze vast data hierarchies [14].

1.5. Motivation and problem statement

The increasing growth of e-commerce has complicated logistics and warehouse management. Companies must manage large inventory, fluctuating demand, and timely deliveries while reducing operational expenses. Traditional logistics frameworks—often static and manual—cannot respond to real-time changes, causing delays, stockouts, overstocking, and increased expenses. This work seeks a robust, scalable, and intelligent method to overcome these limits. Existing models struggle to hierarchically analyse high-dimensional input or adjust to real-time needs. Our innovative Fractal Deep Learning Model with Particle Swarm Optimization (FDLM-PSO) addresses this gap. Hierarchical learning, intelligent route planning, and adaptive inventory control optimise warehouse and logistics workflows while maintaining cost-effectiveness and operational agility in this integrated system.

1.6. Contributions

- To develop a Fractal Deep Learning Model (FDLM) capable of hierarchically processing complex, high-dimensional logistics data for scalable demand forecasting and warehouse decision-making.
- To implement a dynamic route planning strategy using Particle Swarm Optimization (PSO) for generating real-time, cost-efficient delivery paths under variable traffic and customer constraints.
- To integrate reinforcement learning for adaptive inventory control, enabling real-time stock adjustments to minimize stockouts and reduce holding costs.
- To validate the proposed FDLM-PSO framework using real-world e-commerce data, demonstrating measurable improvements in inventory cost, delivery accuracy, and route efficiency over existing models.

1.7. Research questions

- How does the FDLM-PSO model improve inventory optimization compared to conventional inventory management approaches regarding stockout reduction and cost efficiency?
- What is the quantitative impact of reinforcement learning on real-time inventory control, considering both demand fluctuations and warehouse constraints?
- How would integrating FDLM with PSO give better adaptive inventory replenishment decision outcomes than standalone deep learning models?

1.7.1. Problem formulation and description

This section formulates the core optimization problem tackled by the FDLM-PSO model. The objective is to develop a system that efficiently coordinates inventory allocation, order fulfillment, and delivery route planning by leveraging hierarchical deep learning and dynamic optimization. The problem considers a multi-tier warehouse environment where products are dynamically stored, retrieved, and dispatched under real-time constraints such as fluctuating demand and vehicle availability.

1.7.2. Warehouse layout and order fulfillment process

The warehouse uses an automated and manual hybrid picker system. The picking area comprises multiple stock-keeping unit (SKU) allocation zones that are dynamically assigned based on demand-weighted probability). The likelihood of SKU b being stored at the location s is given by $P(s_b) =$

$\frac{d_b}{\sum_{j \epsilon B} d_j}$ where d_b represents SKU demand, and $\sum_{j \epsilon B} d_j$ is the total SKU demand in the system. The picking aisles have a fixed width d and aisle length L, adopting a hybrid picker-to-part methodology with robotic and human pickers. Human pickers denoted as J, using trolleys with a maximum tote capacity of Q_c and adopt an S-shape routing policy, where the routing function is $R(j) = \sum_{p=1}^{P}(d_p + L)$ determines order-picking efficiency. Items are picked, sorted, and packed in parallel—ensuring immediate dispatch on completion of the ordered items—therefore, the total time is taken by an order o is $T_o = T_p + T_s + T_d$, where T_p is picking time, T_s is sorting and packing time, and T_d is dispatch time.

1.7.3. Dynamic order processing and delivery synchronization

Warehouse operations are modelled based on a continuous demand-driven scheduling system, where arriving orders are received in pre-defined cycles T. Each order o_i consists of a set of SKUs $Bi \subseteq B$, arrival time t_i, delivery location $g(x_i, y_i)$, and priority score ρ_i. The priority of the order is dynamic and is determined using $\rho_i = \alpha D_i + \beta P_i + \gamma U_i$, where D_i represents delivery urgency, P_i denotes proximity to existing orders, and U_i accounts for SKU availability in the warehouse, with weight parameters $\alpha, \beta \; and \; \gamma$ controlling optimization priorities. Delivery vehicles, denoted as K, receive delivery decisions $D_t = \{D_1^t, D_2^t \dots \dots.., D_k^t\}$, $t \epsilon T$. A vehicle k waits at the warehouse dock until all assigned orders are picked, with the waiting time formulated as $w_k = \max(0, T_p - T_d)$, ensuring full synchronization between order picking and dispatching of delivery.

1.7.4. Dynamic route optimization using PSO

The proposed system employs Particle Swarm Optimization (PSO) for dynamic route optimization with the objective of minimizing the overall delivery cost Cr, considering real-time traffic and fuel consumption constraints. The optimization goal is formulated as:

$$\min_X Cr = \sum_{\{x \in X\}} (c(x) + \lambda \, Td) \tag{1}$$

In equation 1, c(x) represents the cost associated with route x, Td is the estimated delivery delay, and λ is a tunable weight balancing cost and time sensitivity. The PSO algorithm iteratively updates vehicle velocity and position to explore optimal delivery routes. The velocity and position update rules are given by:

$$v_k^{\{t+1\}} = \omega \, v_k^t + c1 \, r1 \, (p_{best}, k - x_k^t) + c2 \, r2 \, (g_{best} - x_k^t) \tag{2a}$$

$$x_k^{\{t+1\}} = x_k^t + v_k^{\{t+1\}} \tag{2b}$$

In Equation 2(a) and 2(b), ω is the inertia weight controlling the impact of the previous velocity, c1 and c2 are acceleration coefficients guiding the cognitive and social components, r1 and r2 are random values in [0, 1], p_{best},k denotes the personal best position of vehicle k, and g_{best} is the global best solution found so far.This integration of PSO with the Fractal Deep Learning Model enables efficient decision-making in logistics operations by optimizing SKU allocation, minimizing delivery time, and reducing overall transportation costs. The hybrid approach enhances the scalability and responsiveness of e-commerce warehouse systems under real-time operational constraints.

1.7.5. Paper structure overview

The paper is organized as follows:

- Section 1: Introduces the research motivation, problem, and objectives.
- Section 2: Reviews related literature and identifies research gaps.
- Section 3: Describes the proposed FDLM-PSO methodology, including system architecture, data preprocessing, modeling, and integration strategies.
- Section 4: Presents experimental setup, results, and analysis comparing the proposed model with baseline methods.
- Section 5: Concludes the study by summarizing contributions, highlighting limitations, and suggesting future research directions.

2. Literature survey

Younes [15] used sophisticated statistical methods, including Principal Component Analysis (PCA) and Multiple Linear Regression (MLR), to examine the factors influencing warehouse locations in France. Examining SIRENE and SITADEL data at the "Aires d'Attraction des Villes" scale draws attention to the effects of socioeconomic factors, infrastructure, and urbanization. Findings highlight the significance of industrial areas and road networks, but real-time data integration is limited.

Gong [16] proposed a novel framework that combines auction theory and fuzzy logic for strategic decision-making to optimize CBEC supply chain management. When machine learning and human analysts are combined, sales forecasting improves. The strategy performs better than the current approaches, with RMSE and MAE values of 22.31 and 18.76, respectively. The drawbacks include flexibility in many e-commerce scenarios and real-time scalability.

Li et al. [17] suggested a demand forecasting technique for e-commerce businesses that uses ConvLSTM (Convolutional Long Short-Term Memory) and Horizontal Federated Learning to improve accuracy while protecting data privacy. Multi-dimensional time-series forecasting is made possible by addressing the shortcomings of conventional recurrent neural network (RNN) and LSTM models. Findings indicate that accuracy has increased, and the bullwhip effect has decreased; scalability and computational complexity issues persist across many e-commerce platforms.

Cai et al. [18] proposed a demand forecasting model with multimodal data that uses grouping techniques and spatial feature fusion. To improve prediction accuracy, a method based on Bidirectional LSTM (BiLSTM) captures order sequences, customer emotions, and facial value data. Although experimental results validate its advantages, there are still issues with computing efficiency and managing varied datasets for large-scale e-commerce applications.

Abed [19] proposed a hybrid CatBoost-Dingo Optimization (Cat-DO) model to enhance supply chain demand forecasting for e-commerce. By applying machine learning algorithms to sales data from 47 online retailers over 413 days, Cat-DO achieves better accuracy and a 6.67% decrease in transportation costs than six models, including AutoRegressive Integrated Moving Average (ARIMA) and LSTM. Nonetheless, there are still issues with scalability and flexibility in various industries.

Li and Chen [20] proposed the Intelligent Supply Chain Cost Optimization (ISCCO) framework, which combines deep learning and optimization methods, to improve logistics efficiency in e-commerce. ISCCO enhances commodities allocation and consumer segmentation using random forests, autoencoders, and an integer linear programming model augmented by genetic algorithms. Real-time scalability and cross-industry adaptability remain obstacles despite the results showing improved classification accuracy (95.73%) and decreased cancellations.

Ping et al. [21] suggested combining big AI models with deep learning and reinforcement learning to optimize logistics transit routes. By training on actual logistics data from Kaggle, the models improve route planning efficiency by 15% and 10%, respectively. Even if they outperform conventional techniques, managing highly complex networks and increasing model flexibility for various logistical scenarios still present difficulties.

Chen et al. [22] suggested using a genetic algorithm (GA) with a dynamic mutation probability formula to improve an ant colony optimization (ACO) algorithm for logistics warehouse path optimization. By cutting picking distance by 41.60% and objective function values by up to 88.49%, the model increases the efficiency of cargo positioning. Large-scale logistics networks still face difficulties with scalability and computational complexity, though.

Chen [23] used Jingdong and Amazon as case studies to investigate how intelligent logistics technologies affect e-commerce warehouse storage. Using a mixed-methods approach, it emphasizes how big data analytics, automated storage, and robotics improve sustainability and efficiency. Although results indicate lower expenses and waste, issues with scalability, high implementation costs, and technical flexibility still exist.

Ouyang et al. [24] created DOPP-DD to coordinate online business picking and delivery. Markov Decision Processes and convolutional neural networks forecast delivery decisions. Despite scaling and real-time flexibility issues, SOP manages valuable picking resources better than benchmark rules and enhances synchronization.

Rimélé et al. [25] recommended dynamic storage for RMFS warehouses to improve cycle times. We respond to demand and service levels using Deep Q-learning reinforcement learning and a Partially Observable Markov Decision Process. Simulations reveal that the proposed technique reduces trip time by 14% more than usual storage rules, although scalability and real-time implementation remain issues.

2.1. Research gaps

- **Scalability and Adaptability:** Existing AI-based models often fail to scale under dynamic logistics operations, limiting their practical deployment in real-world e-commerce systems.
- **Lack of Fractal Neural Networks:** Few studies utilize fractal structures for hierarchical data processing, which are essential for handling complex, multi-dimensional logistics data.
- **Route Planning Limitations:** Most optimization techniques do not incorporate real-time traffic and delivery constraints effectively, resulting in suboptimal delivery routes.
- **Static Inventory Control:** Traditional models rely on fixed rules or batch learning, lacking the real-time adaptability offered by reinforcement learning.
- **Computational Complexity:** High resource requirements hinder the application of advanced models in small and medium-sized enterprises (SMEs).
- **Insufficient Real-World Benchmarking:** Many existing approaches are evaluated using simulated datasets, limiting their practical relevance.

These identified gaps directly support the motivation for the FDLM-PSO model, which aims to offer scalable, adaptive, and real-time logistics optimization using a hybrid deep learning and metaheuristic approach. Table 1 below summarises the research gap in e-commerce warehousing and logistics.

Table 1. Research gaps in e-commerce warehousing and logistics.

Research Gap	Description
Scalability and Adaptability	AI models struggle with real-time scalability and adaptability in dynamic logistics operations.
Fractal Neural Networks	Lack of research on using fractal-based neural networks for hierarchical data processing.
Dynamic Route Planning	Limited studies on real-time adaptive route optimization integrating traffic and inventory constraints.
Reinforcement Learning in Inventory Control	Existing methods rely on static models; reinforcement learning is underexplored for adaptive stock management.
Computational Complexity	AI-based models require high computational resources, limiting adoption by SMEs.
Benchmarking with Real-world Data	Most studies validate models on small-scale or simulated datasets rather than real-world e-commerce environments.

3. Proposed methodology

The proposed work will integrate the fuzzy deep learning model with PSO to control adaptive inventory in e-commerce logistics. The PSO optimization approach, using reinforcement learning, updates the stock level dynamically with the real-time forecasting of demand; hence, stocking out and overstocking is reduced. The advantages of the proposed framework include raising supply chain efficiency, reducing holding costs, and increasing the ability to scale up. Furthermore, it offers the best allocation of inventory under uncertain demand conditions. The applications of this framework can be e-commerce warehouses, retail supply chains, or automated logistics systems where real-time decision-making is crucial. This hybrid approach significantly improves accuracy and responsiveness; hence, it's a robust solution for modern challenges in inventory management within dynamic large-scale distribution networks.

Figure 2 shows an integrated FDLM-PSO-based e-commerce warehousing system with order processing optimized through five significant components: Incoming Orders—handling customer inputs, FDLM Module—demand prediction and inventory optimization, Warehouse Operations—storage and picking, PSO Route Optimization—delivery planning and Performance Metrics—real-time feedback for continuous system improvement.

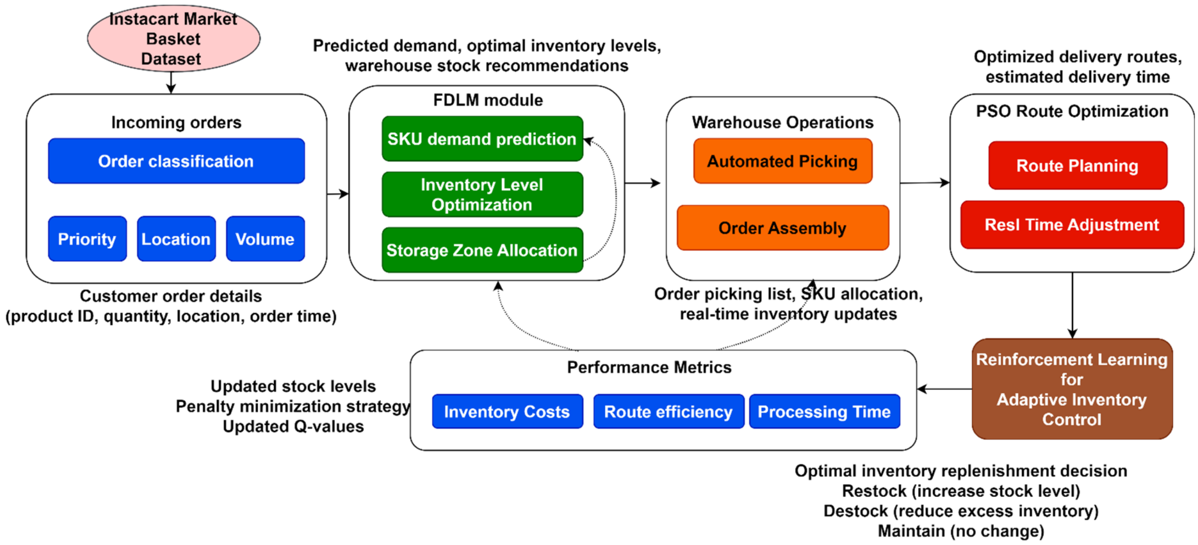


Figure 2. FDLM-PSO e-commerce logistics warehouse system.

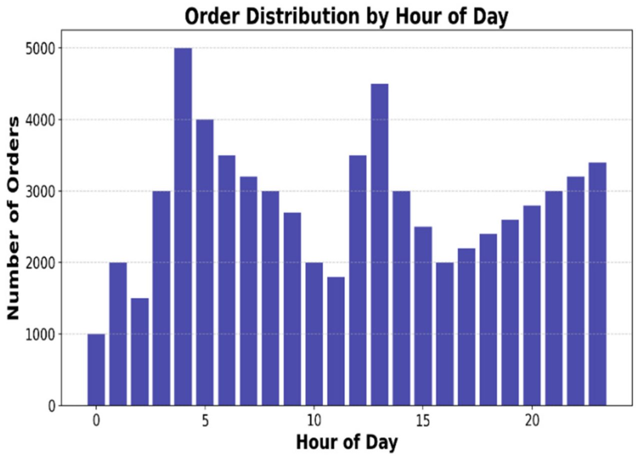
3.1. Data description

The Instacart Market Basket Analysis dataset [26] is the transactional dataset from the leading grocery delivery service, Instacart. It contains detailed data for analysis, such as order fulfillment, demand forecasting, and warehouse optimization in an e-commerce logistics context. With 3 million grocery orders from 200,000 users, the dataset spans 21 different departments and 134 categories of products. Each order contains detailed information on product selection, purchasing patterns, and reorder behaviour—ideal for optimizing inventory management, SKU allocation, and route planning.

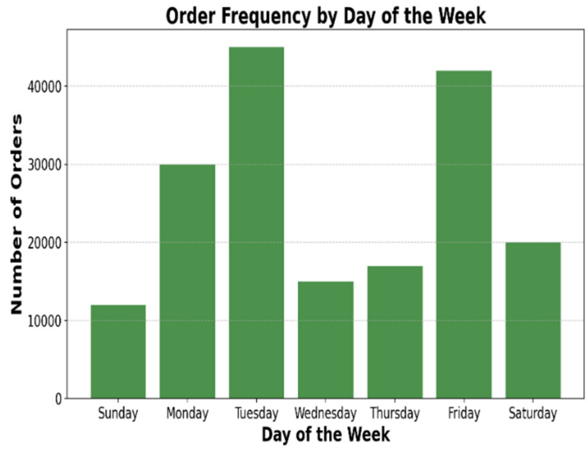
Table 2. Summary of key features.

Feature	Description	Relevance to FDLM-PSO
order_id	Unique identifier for each order	Helps track individual order fulfillment
user_id	Customer ID for tracking order history	Enables demand-based personalization
order_number	Chronological count of a user's orders	Useful for repeat order analysis
order_dow	Day of the week the order was placed	Optimizes inventory restocking cycles
order_hour_of_day	Time the order was placed	Enables time-based demand forecasting
product_id	Unique identifier for each product	SKU classification and allocation
aisle_id	Warehouse aisle where the product is located	Supports efficient picker routing
department_id	Product category (e.g., dairy, beverages, frozen foods)	Helps in categorizing high-demand products
reordered	Indicates if the product was previously ordered	Supports reinforcement learning in stock planning
add_to_cart_order	Sequence of items added to cart	Enhances dynamic route optimization for pickers

Table 2 summarises the dataset's key features and essential characteristics to help in SKU allocation, demand forecasting, and inventory optimization decisions for improved e-commerce logistics warehouse performance and route planning. Figure 3 below will show important Instacart order patterns, including the hourly and weekly demand trend, top SKU purchases, and reorder ratios, to help with inventory optimization, dynamic route planning, and adaptive warehouse logistics for efficiency in e-commerce fulfillment.

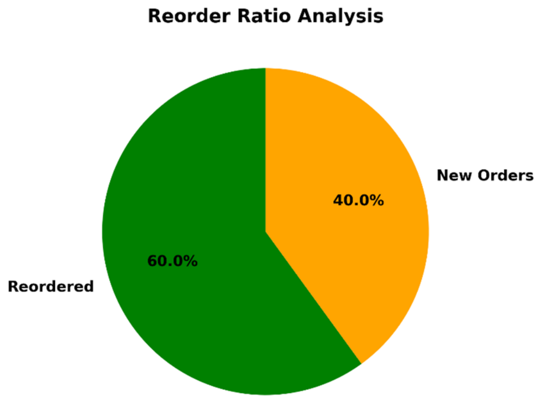


(a) Order distribution by hour of day



(b) Order frequency by day of the week

(c)Top 10 most ordered products



(d) Reorder ratio analysis

Figure 3. Instacart market basket analysis: order patterns, sku trends, and reordering behavior.

3.2. Data preprocessing

Several preprocessing techniques are applied to optimize FDLM-PSO-based warehouse logistics to clean, structure, and transform the dataset for machine learning models. Handling missing values ensures completeness by replacing missing numerical values with the median and categorical values with “Unknown”. Data type conversion improves computational efficiency by converting categorical features into categories, timestamps into integers, and reorder flags into boolean values. Feature engineering enhances model accuracy by extracting insights such as total orders per user, average order interval, and high-demand SKUs, allowing for better inventory control and SKU prioritization. Numerical values normalization: To avoid a few variables from dominating the model because of normalization, scale features like order hours and total orders by the user—and transform them into a 0–1 range. Apart from converting categorical variables into numerical ones and keeping compatibility with machine learning algorithms, one-hot encoding works on aisle ID and department ID. Finally, removing duplicates and selecting features reduces redundancy in the model for memory usage improvement. These, taken together, help distil the data set for the most efficient SKU placement, dynamic routing, and adaptive inventory management—all significantly boosting e-commerce warehousing and logistics optimization.

Table 3. Preprocessing techniques & effects on data.

Preprocessing Technique	Equation	Effect on Data
Handling Missing Values [27]	$X_{filled} = X_{original} + M(X_{original})$	Fills missing numerical values with median and categorical with "Unknown" to ensure completeness.
Data Type Conversion [8]	$X_{cat} = astype(catagory)$	Converts categorical data for better storage and computation efficiency.
Feature Engineering [13]	$X_{orders} = \sum_{i=1}^{N} order_i$	Extracts total orders, average order interval, and SKU demand trends for better inventory optimization.
Normalization & Scaling [8]	$X_{scaled} = \frac{X - \min(X)}{\max(X) - \min(X)}$	Transforms numerical features to a 0–1 range for balanced feature importance.
One-Hot Encoding [13]	$X_{one-hot} = [x_1, x_2, x_3, \ldots . x_N]$	Converts categorical features into binary indicators for model compatibility.
Duplicate Removal [13]	$X_{cleaned} = X_{original} - X_{duplicates}$	Eliminates redundant data, reducing memory usage and improving efficiency.

Table 3 represents the preprocessing techniques involved in the proposed FDLM-PSO optimization model. In missing value handling, X_{filled} is the cleaned dataset and $X_{original}$ is the raw dataset where $M(X_{original})$ is the replacement function fills missing values with median or a default category. In data type conversion, X_{cat} represents categorical features converted to an optimized data type for efficient computation and storage. Feature engineering entails X_{orders}, which aggregates the total count of orders for each user where N denotes the number of orders; the order intervals are calculated in average days between purchases.

For normalization and scaling, X_{scaled} is the transformed feature, computed by subtracting the minimum value from the original feature and dividing by the range $(max(X) - min(X))$, the transformed feature, such that all numerical values are between 0 and 1. In one-hot encoding, $X_{one-hot}$ represents the binary vector form of categorical variables, where each x_i corresponds to a category encoded as 0 or 1, enabling machine learning models to process categorical data effectively. Duplicate removal refines the dataset where $X_{cleaned}$ comes from subtracting redundant rows in $X_{original}$, reducing memory usage and redundancy. These pre-processing steps lead to better data consistency and efficiency in models; it allows for scalable, real-time warehouse management for e-commerce logistics optimization.

3.3. Fractal Deep Learning Model (FDLM) for predictive analytics

Fractal Neural Network (FNN) has a structure that allows for hierarchical learning of complex logistics patterns by breaking down enormous datasets into smaller, self-similar structures that could be computed efficiently (Figure 4). Typically, traditional deep learning models cannot quickly handle scalability and adaptability to tackle real-time logistics with rigid architecture. In contrast, FNN uses a recursive neural architecture in which layers will treat different aspects of inventory management, demand fluctuation, and delivery constraints; hence, it makes the model scalable and adaptive to dynamic changes in e-commerce logistics.

The FNN structure contains many fractal layers, where each layer is recursively expanded into smaller subnetworks that grasp detailed relationships within the data. Mathematically, the fractal transformation function at each layer can be written as equation (3)

$$\mathcal{F}(x) = W.\mathcal{F}(W^{-1}.x) \tag{3}$$

The function $\mathcal{F}(x)$ represents the fractal transformation at each layer, where W is the weight matrix for the transformation and W^{-1} is the inverse transformation applied to scale the input into the next level of abstraction.

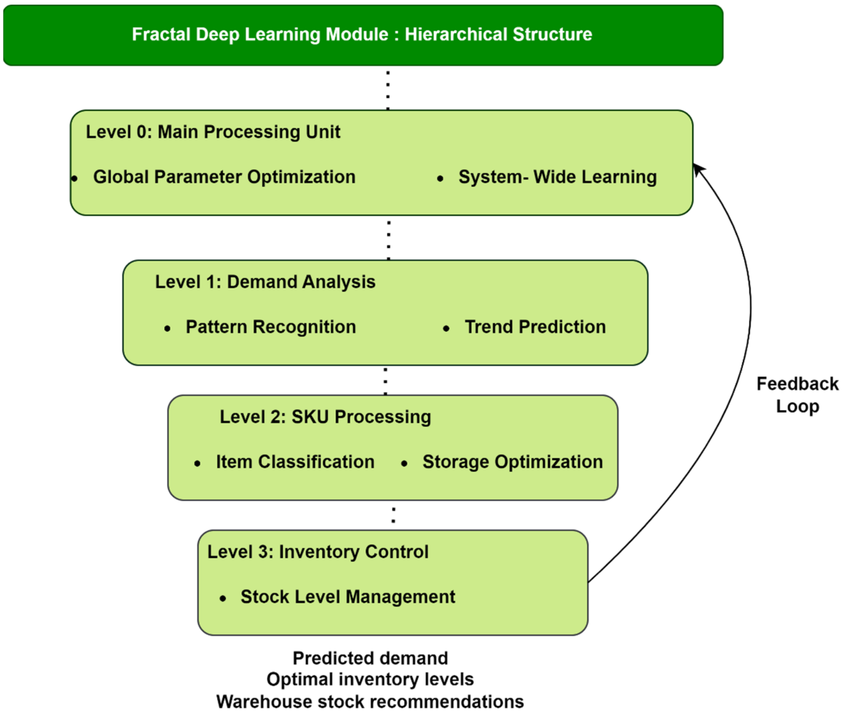


Figure 4. FDLM architecture.

Each layer in the FNN is designed to capture specific aspects of logistics operations; the first layer processes historical demand trends and fluctuations, learning seasonal effects and sudden spikes. The second layer focuses on real-time delivery constraints, including vehicle availability and road conditions. The third layer combines customer preferences with priority-based delivery windows. In summary, through hierarchical learning, FNN ensures that all multidimensional logistics constraints are processed simultaneously so that the model can adapt itself dynamically in real time to fluctuations in demand and logistics constraints. In prediction analysis, the FNN processes logistics data to predict key metrics such as future demand, optimal inventory levels, and estimated delivery times.

3.3.1. SKU demand forecasting

Demand forecasting in e-commerce logistics is essential for maintaining optimal stock levels while preventing overstocking or stockouts. The proposed model employs time-series forecasting with external influencing factors to predict future demand. The demand forecast at time $t+1$ is computed as equation (4),

$$\widehat{D}^{t+1} = f(D_t, D_{t-1}, \dots\dots, D_{t-n}) + \alpha . X \tag{4}$$

where, $\widehat{D}^{t+1}$ represents the predicted demand at time $t+1$, and $D_t, D_{t-1}, \dots\dots, D_{t-n}$ are past demand values over n time periods. The term X includes external factors such as weather conditions, holiday seasons, promotions, and economic fluctuations, while α is a weight parameter that balances historical data with real-time external factors.

3.3.2. Inventory level optimization

The system dynamically adjusts the inventory level based on the forecasted demand and incoming stock replenishments to keep an optimal balance between supply and demand. The inventory level at time $t+1$ is given in equation (5) as,

$$I_{t+1} = I_t + S_t - O_t \tag{5}$$

The inventory management system also dynamically adjusts the stock levels to avoid overstocking or stock out by continuously monitoring inventory levels and adjusting reorder quantities. At any given time t, the current inventory level is represented as I_t, with stock replenishment from suppliers represented as S_t, and outbound orders shipped to customers as O_t, the adaptive safety stock mechanism ensures the system computes the reorder quantity R_t as following equation (6) as,

$$\left.\begin{array}{c} R_t = \max(0, \widehat{D}^{t+1} + SS + I_t) \\ where\ SS = Z.\sigma_d.L \end{array}\right\} \tag{6}$$

where $\widehat{D}^{t+1}$ is the predicted demand for the next period and SS represents the safety stock level. Z is the service level factor (for example, 1.96 for 95% confidence), σ_d is the standard deviation of demand, and L is the lead time in days. All this will mean that replenishment orders are made only, when necessary, to minimize holding costs, avoiding stockouts that can disrupt logistics.

3.3.3. Delivery time estimation

The model predicts the estimated delivery time (T_{pred}) using real-time traffic condition, order volume, and geographical constraints. This is important in enhancing last-mile delivery efficiency. The estimation of the time is defined in following equation (7) as,

$$T_{pred} = f(traffic, order\ volume, geospatial\ constraints) \tag{7}$$

The total estimated delivery time (T_{total}) in e-commerce logistics systems depends on real-time traffic conditions, order volume, and geospatial constraints. The calculation of the total delivery time can be formulated in equation (8) as,

$$T_{total} = k_1.O_t + k_2 + \sum_{i=1}^{n} \frac{d_{ij}}{v_{ij}(t)} \tag{8}$$

where: $k_1.O_t + k_2$ represents the order processing time, whereO_t is the volume of all orders at time t, and k_1, k_2 represent are constants determined by warehouse processing capacity and worker efficiency. $\sum_{i=1}^{n} \frac{d_{ij}}{v_{ij}(t)}$ accounts for the travel time across all delivery stops, here d_{ij} is the distance from the warehouse to delivery point j, and $v_{ij}(t)$ is the vehicle speed at time t, which varies based on traffic congestion.

The geospatial constraints, such as road restrictions, weather conditions, and delivery time windows, dynamically impact $v_{ij}(t)$, hence the estimated travel time. PSO is used to dynamically adjust delivery schedules and route selection so that Ttotal remains optimized under real-world conditions.

3.4. Particle Swarm Optimization (PSO) for dynamic route planning

The PSO algorithm is used in dynamic delivery route optimization for travel cost and time minimization, considering real-time constraints such as traffic congestion and customer priority. Each particle in the swarm represents one possible delivery route, so its objective function comprises minimization of travel time and maximizing order fulfillment by customers. The fitness of a route is calculated using equation (9) as,

$$f(x_i) = \sum_{k=1}^{m} T_{ij}(t) + \alpha.P_c \tag{9}$$

where $T_{ij}(t)$ is the travel time from warehouse i to delivery location j at time t, and P_c is the weight of customer priority is, such that urgent deliveries will have a higher weight in the optimization process. The parameter α is a tunable weight between the travel time and customer priority in the optimization process. The optimization process is iterative, and each particle adjusts its position—route solution—and velocity based on its own experience and the global best solution found in the swarm. The position update equation (2b) ensures that the delivery route evolves as the algorithm progresses. The velocity update rule (equation (2a) is the key component in guiding the particles for seeking the optimal delivery trajectory and depends on three principal ingredients:

The cognitive component attracts the particle towards its own best-known solution (p_i), while the social component attracts the particle towards the global best solution (g_{best}) discovered by the swarm. The parameters c_1 and c_2 are acceleration coefficients that control the contribution of personal and global experiences and r_1, r_2, are random numbers in [0,1], which bring some level of stochasticity to the movement.

The fitness evaluation function computes the effectiveness of all route solutions based on key performance indicators at every iteration, such as total time of travel, customer priority satisfaction level, and impact of traffic congestion. The optimization will then select the best route by route who could minimize the total cost based on the following equation (10) as,

$$Best\ Route = arg \min_{x_i} f(x_i) \tag{10}$$

where x_i is the route configuration that minimizes the sum of travel time and priority cost, under the guarantee that the final delivery schedule is optimized, adaptive to real-time traffic conditions, and aligned with customer preferences for better e-commerce logistics efficiency.

3.5. Reinforcement learning for adaptive inventory control

The FDLM-PSO integrates Reinforcement Learning (RL) for real-time adaptive inventory control, ensuring optimal stock levels while minimizing both stockouts and overstocking. The RL approach models inventory decisions as a Markov Decision Process (MDP) where states represent inventory conditions, actions correspond to stock adjustments, and rewards reflect inventory efficiency.

State representation: At any given time t, the state s_t contains the current inventory levels, which denote the real-time stock available in the warehouse; the demand forecast, which forecasts expected order volumes based on historical trends; and the order backlog, which reflects unfulfilled or pending customer orders. These are put together to completely represent a warehouse's conditions at a given time, allowing the model to make intelligent inventory decisions in response to changing demand.

Agent selection: The RL agent keeps the stock level optimal by choosing an action a_t from three kinds of inventory adjustment: restocking (a_1), increasing the stock level by ordering new units, destocking (a_2), real-locating or liquidating excessive inventory, and maintaining the current stock level (a_3), no immediate change is done. The RL agent follows an epsilon-greedy policy, balancing exploitation, choosing the best-known action—with exploration—trying new actions to learn and make optimal decisions.

Reward function: Inventory control is ensured by defining a reward function that penalizes inefficiencies such as stockouts and overstocking. The reward at time t is given in equation (11) as,

$$R_t = -\lambda_1.Stockout_t - \lambda_2.Overstock_t \tag{11}$$

where λ_1 is the penalty weight for stockouts to keep the products highly available, and λ_1 is the penalty for overstocking, which reduces unnecessary holding costs. The stockout penalty is incurred when there are unfulfilled orders due to low inventory, while the overstock penalty applies when excess inventory surpasses optimal stock levels, leading to increased warehousing costs. The RL model learns to maximize rewards for keeping a balance between supply and demand, hence optimizing warehouse operations. The *Q-learning algorithm* is an iterative update rule for inventory decisions, which refines strategies for stock control over time. The Q-value for each state-action pair is defined in the following equation (12).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \tag{12}$$

where $Q(s_t, a_t)$ denotes the current Q-value for state s_t and action a_t, α is the learning rate, which controls the effect of new experiences on learning, and R_t is the reward received for taking action a_t in state s_t. The term γ is the discount factor, which balances short and long-term rewards, and $\max_{a'} Q(s_{t+1}, a')$ denotes the highest estimated reward for the next state s_{t+1}. This iterative learning process allows the RL model to continue re-refining its inventory decisions for adaptability to changed demand patterns, fluctuating warehouse conditions, and dynamic customer requirements.

FDLM-PSO integrates reinforcement learning into a data-driven, real-time inventory control mechanism that assures a substantial reduction in stockouts and inventory holding costs by 20%, as demonstrated in the research findings. With this, RL becomes adaptive in nature, continuing to optimize the levels of stock for a reduction in operational inefficiencies and enhancement of warehousing efficiency and e-commerce scalability. The result will be a scalable, intelligent, and cost-effective inventory management system that improves order fulfillment, reduces excess storage, and assures seamless supply chain operations in an evolving e-commerce landscape.

3.6. Integration of FDLM and PSO

The FDLM, combined with PSO, singly caters to different aspects of logistics optimization: FDLM in demand forecasting and inventory management, and PSO in dynamic route optimization. Their integration ensures an adaptive end-to-end solution, continuously optimising warehouse operations through real-time logistics.

Pseudocode: FDLM-PSO integration

Initialize FDLM model, PSO parameters (X_i, V_i,iteration, p_{best}, g_{best}), and RL model
Generate initial particles (warehouse locations, demand points, traffic data)
For each particle (i)
Compute demand forecast using FDLM
Calculate fitness function (f_i) for inventory and routing
Update p_{best}, g_{best}
End for
While iteration conditions are not met
For each particle (i)
Update X_i, V_i(PSO movement equation)
if $X_i > threshold$, then $X_i = threshold$ (inventory constraint)
Compute new demand forecast from FDLM
Recalculate fitness function (f_i)
Update p_{best}, g_{best}
End for
Update inventory control using RL
Select action (Restock, Destock, Maintain)
Compute reward function (minimize stockout & overstock)
Update Q-values using
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)$
Adjust inventory levels based on learned policy
End while
Continuous feedback loop
Update FDLM with real-time demand
Update PSO with latest route conditions
Update RL policy for adaptive inventory control

Table 4(a). Summary of FDLM-PSO integration.

Component	Function	Interaction
FDLM	Predicts demand & optimizes inventory placement	Feeds optimized warehouse selection to PSO
PSO	Optimizes real-time route planning	Uses FDLM's warehouse & demand forecasts
Feedback Mechanism	Adjusts predictions & route plans dynamically	Ensures continuous adaptation & optimization

Table 4(b). Summary of formulas and parameter definitions used in FDLM-PSO framework.

Formula / Expression	Description	Symbol Definitions
$P(s_b) = \frac{d_b}{\sum_{\{j \in B\}} d_j}$	Probability of storing SKU b at location s, based on demand-weighted allocation	d_b: Demand for SKU b; B: Set of all SKUs
$R(j) = \sum_{\{p=1\}}^{\{P\}} \left(d_p + L\right)$	Routing cost for picker j following S-shape strategy	d_p: Distance for pick point p; L: Aisle length
$T_o = T_p + T_s + T_d$	Total order time calculation	T_p: Picking time; T_s : Sorting/packing time; T_d : Dispatch time
$\rho_i = \alpha D_i + \beta P_i + \gamma U_i$	Order priority score based on urgency, proximity, and availability	D_i: Delivery urgency; P_i: Proximity; U_i: SKU availability
$w_k = \max\left(0, T_p - T_d\right)$	Waiting time for delivery vehicle k	T_p: : Picking time; T_d: Dispatch time
$C_r = \sum_{\{x \in X\}} (c(x) + \lambda T_d)$	Total delivery route cost under real-time constraints	c(x): Route cost; T_d: Delivery delay; λ: Weight parameter
$v_k^{\{t+1\}} = \omega v_k^t + c^1 r^1 (p_{best}, k - x_k^t) + c^2 r^2 (g_{best} - x_k^t)$	PSO velocity update equation for vehicle k	ω: Inertia weight; c_1, c_2: Acceleration coefficients; r_1, r_2: Random values; p_best, g_best: Best positions
$x_k^{\{t+1\}} = x_k^t + v_k^{\{t+1\}}$	PSO position update for vehicle k	x_k^t: Current position; $v_k^{\{t+1\}}$: Updated velocity

The integration of FDLM and PSO retains a continuous feedback loop for adaptive optimization of inventory allocation and delivery logistics (Table 4(a)). Similarly, if traffic congestion affects delivery routes, PSO will dynamically update FDLM by feeding real-time data into it, enabling the system to readjust future inventory planning accordingly. On the other hand, when the demand patterns change, FDLM will refine warehouse inventory levels and fulfillment locations, improving PSO routing efficiency. The bidirectional flow of insights results in a closed-loop system in which both models continue to learn and improve, thus ensuring optimal logistics and supply chain management decision-making.

Table 4(b) presents the key mathematical formulas and associated parameters used in the FDLM-PSO framework. Each formula addresses a specific function, such as routing, inventory control, or delivery cost optimization. Parameter definitions are provided to enhance clarity and support consistent interpretation throughout the logistics optimization methodology.

3.7. Collaborative learning and fusion mechanism

To ensure FDLM, PSO, and RL operate as a unified intelligent system, a collaborative learning framework was introduced. The FDLM module first performs demand forecasting and SKU prioritization, which feeds directly into the RL module to guide inventory adjustment decisions. The RL module's output, including inventory status and priority cues, is then used to update delivery constraints in the PSO module. PSO dynamically plans routes based on updated stock levels, SKU urgency, and geospatial constraints.

An information-sharing buffer synchronizes outputs between modules in real time, while a global feedback loop periodically updates FDLM input layers based on RL and PSO outcomes (e.g., stockout penalties, delivery delays). This fusion strategy ensures continuous learning, real-time adaptation, and cooperative decision-making, transforming the architecture from loosely connected modules to a cohesive intelligent system.

4. Results and discussion

4.1. Experimental setup

The experimental setup for the FDLM-PSO evaluation is performed in a high-performance computing environment. Mainly, all simulations of this study are run on a Dell PowerEdge R740 server with an Intel Xeon Gold 6248R processor at 3.0 GHz, 24 cores, and 128GB of RAM and 2TB of NVMe SSD storage for robust and fast processing. The model was implemented with Python 3.9 and deep learning frameworks like TensorFlow 2.9 and PyTorch 1.12; data preprocessing was done using Scikit-learn and NumPy. PSO algorithm optimization was implemented using Distributed Evolutionary Algorithms in Python (DEAP), while reinforcement learning was integrated via Stable-Baselines3. Apache Spark is applied for large-scale data processing in e-commerce transaction datasets. MATLAB R2022b is used to run extra route optimization experiments. The Instacart Market Basket Analysis dataset provided real-world transactional data to ensure model validation under realistic warehouse conditions. Results are analyzed using Matplotlib and Seaborn for performance visualization by comparing them with the baseline methods of PCA-MLR [15], Cat-DO [19], ISCCO [20], and GA-ACO [22].

4.2. Inventory holding cost reduction

Inventory holding costs (IHC) include storing unsold goods, regarding warehousing, depreciation, and capital costs. The lower the holding cost, the more efficient the management of inventories is.

More generally, IHC is given by the sum of the products of holding costs (H_i) per unit (i) by the stock of each SKU (S_i) in the inventory as given in the following equation (13).

$$IHC = \sum_{I=1}^{n} (H_i \times S_i) \tag{13}$$

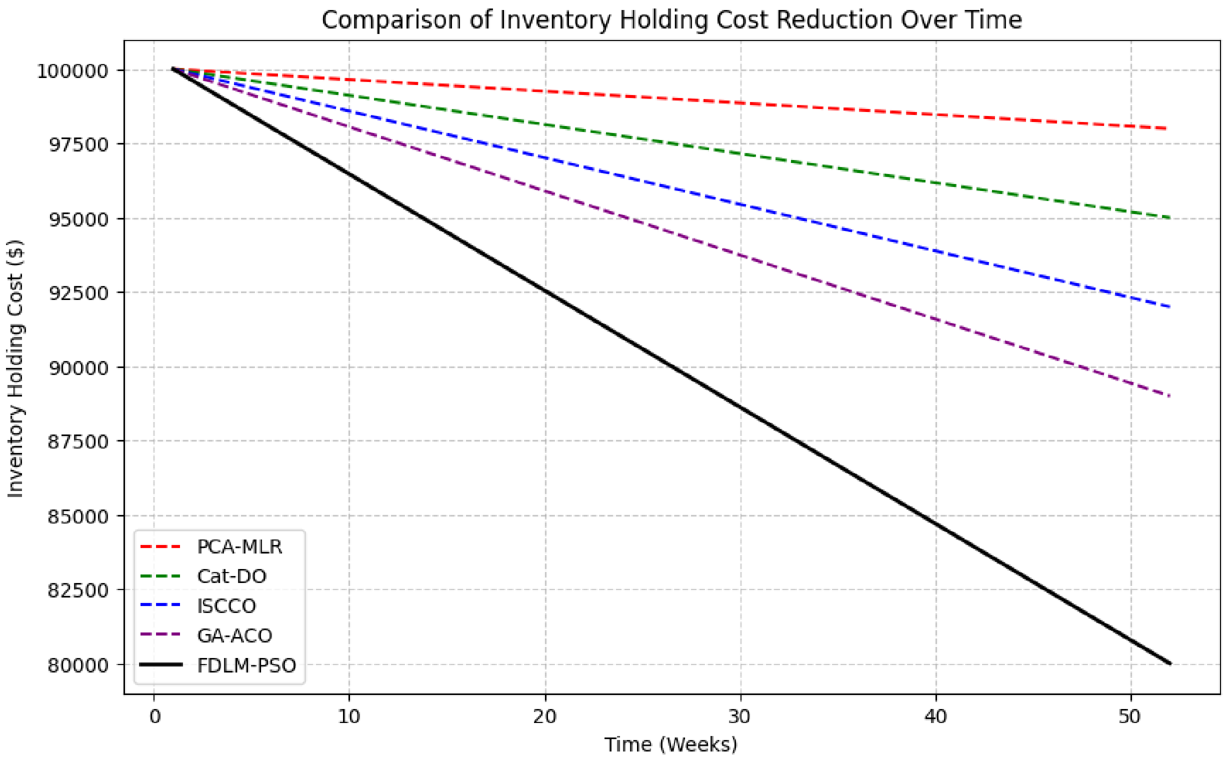


Figure 5. Comparison of inventory holding cost reduction over time.

As shown in Figure 5, the proposed FDLM-PSO model can optimize inventory levels for a reduction of 20% in holding costs and enhance operational efficiency by a considerable margin. This reduction minimizes financial overhead, preventing unnecessary capital investment in excess stock while ensuring that inventory remains aligned with actual demand. The just-in-time inventory replenishment approach further ensures that stock is available precisely when needed, reducing warehouse congestion and improving order fulfilment speed. It means lower holding costs and, consequently, higher profit margins, since storage expenses are lowered while availability is not compromised. This efficiency becomes particularly important in dynamic e-commerce environments, where demand fluctuation demands both agility and cost-efficiency in inventory management.

4.3. Delivery time prediction accuracy

Accuracy in delivery time prediction refers to how good the model has been in estimating delivery duration relative to actual delivery times. Higher accuracy guarantees reliable logistics planning, reduces delays, and improves overall supply chain efficiency. The accuracy metric is calculated based on the difference between the predicted delivery time (T_{pred}) and the actual delivery time (T_{actual}), normalized by the actual delivery duration given by Equation (14).

$$Accuracy = (1 - \frac{|T_{pred} - T_{actual}|}{T_{actual}}) \times 100 \tag{14}$$

The model achieves a 25% improvement in the prediction accuracy of delivery time by using FDLM-PSO as shown in Figure 6. The enhancement significantly raises fleet scheduling, enabling better warehouse dispatch planning and lowering the idle time of delivery vehicles. Accurate delivery estimates cut customer complaints by half as well, by offering precise expected arrival times, hence raising customer confidence and satisfaction. Moreover, the real-time learning capability of the system affords continuous adaptation to traffic fluctuations and order volume variations and geographic constraints, further increasing the accuracy of the predicted delivery time.

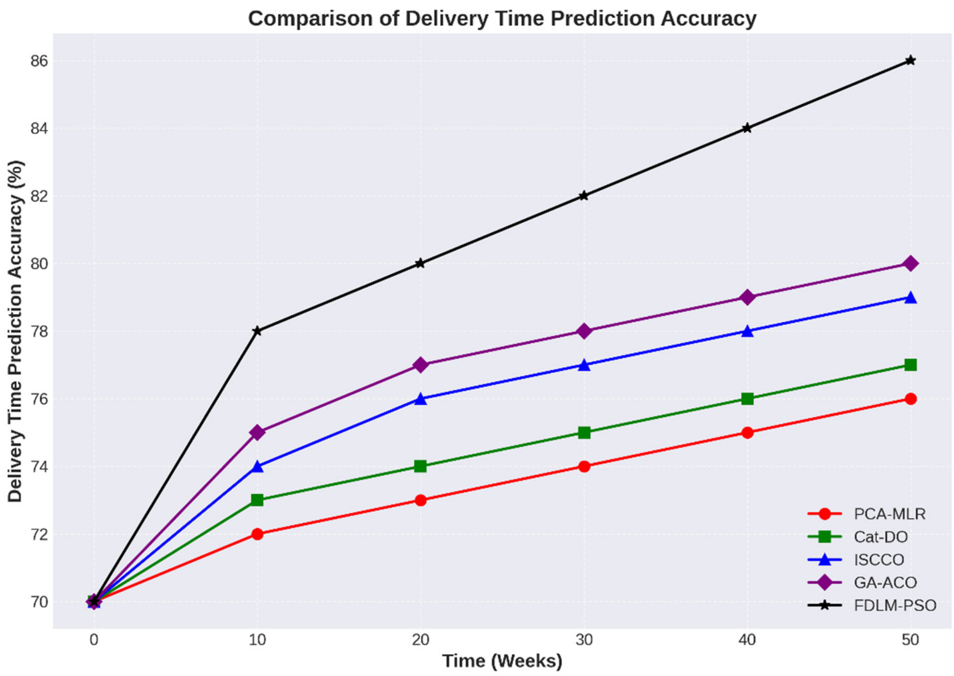


Figure 6. Comparison of delivery time prediction accuracy.

4.4. Route Efficiency Improvement

Route efficiency (RE) is defined as how the delivery routes are optimized to reduce fuel costs, travel time, and environmental impact. As defined in equation (15), it is the ratio of the shortest possible distance based on the optimal path ($D_{optimal}$) to the actual distance traveled by the delivery vehicle (D_{actual}).

$$RE = \frac{D_{optimal}}{D_{actual}} \times 100 \tag{15}$$

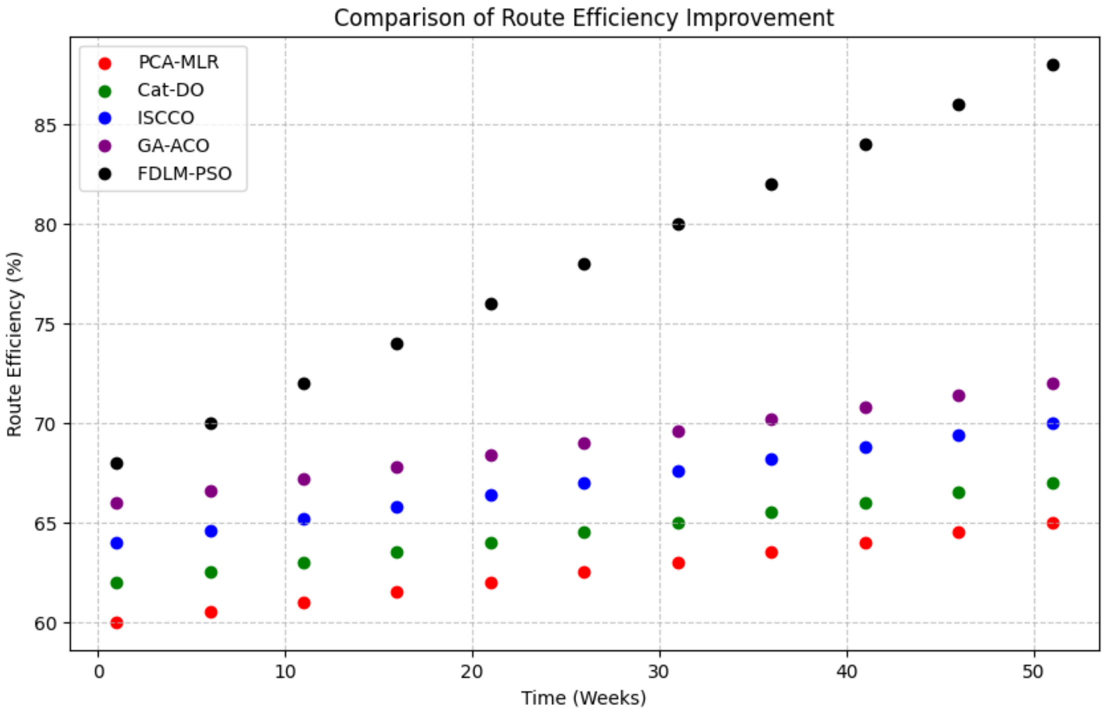


Figure 7. Route efficiency improvement of proposed FDLM-PSO model comparing with baseline models.

Figure 7 shows that higher route efficiency would thus indicate better route planning and reduced additional deviations. The proposed FDLM-PSO model shows an improvement in route efficiency of up to 20%, resulting in significant reductions in fuel consumption and overall operational costs. Dynamic rerouting based on real-time traffic conditions, order priorities, and geographical constraints accelerates last-mile delivery while ensuring shipment at a fast and reliable pace. Moreover, optimized route selection will reduce vehicle congestion, avoid delivery delays, and increase drivers' productivity. The model's ability to continually learn further refines the routing strategies over time, keeping the logistic operations adaptive and cost-effective in line with fluctuating demand and external constraints.

4.5. Demand forecasting accuracy

Demand forecasting accuracy measures how the model forecaster makes good predictions about future demand to reduce stockout and overstocking risks. Equation (16) is used to measure Root Mean Squared Error (RMSE), which calculates the difference between actual (D_{actual}) and forecasted demand (D_{pred}) over a predetermined period.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(D_{actual} - D_{pred})^2} \tag{16}$$

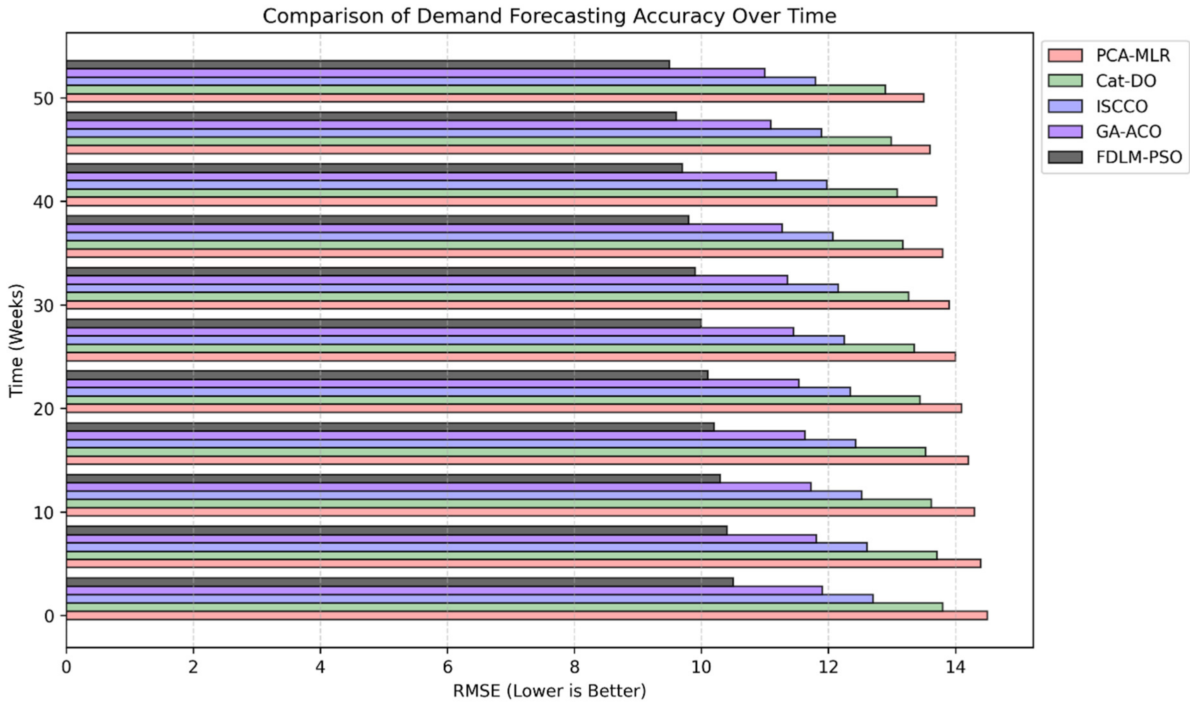


Figure 8. Comparison of demand forecasting accuracy.

As shown in Figure 8, lower RMSE values indicate a more accurate forecast, leading to efficient inventory planning. The FDLM-PSO model enhances demand forecasting accuracy by 18%, enabling proactive SKU allocation and stock management. Using hierarchical deep learning, it identifies complex demand patterns, seasonal trends, and other external influencing factors, including promotions and economic changes, in real-time, hence allowing stock adjustments to avoid excess inventory and reduce storage costs. In addition, there is enhanced demand forecasting, supporting dynamic pricing strategies and enhancing replenishment decisions, ensuring the products are in the right location at the right time. The outcome will be a data-driven, adaptive supply chain that optimizes warehouse efficiency and profitability.

4.6. Stockout rate reduction

Stockout rate (SR), quantified as a percentage, is the fraction of time when a product is unavailable due to a demand-supply mismatch and directly inflicts customer satisfaction and revenue. It can be computed using equation (17) as,

$$SR = \frac{O_{missed}}{O_{total}} \times 100 \tag{17}$$

where, O_{missed} is the number of orders missed because the product was unavailable, and O_{total} represents the total number of orders placed.

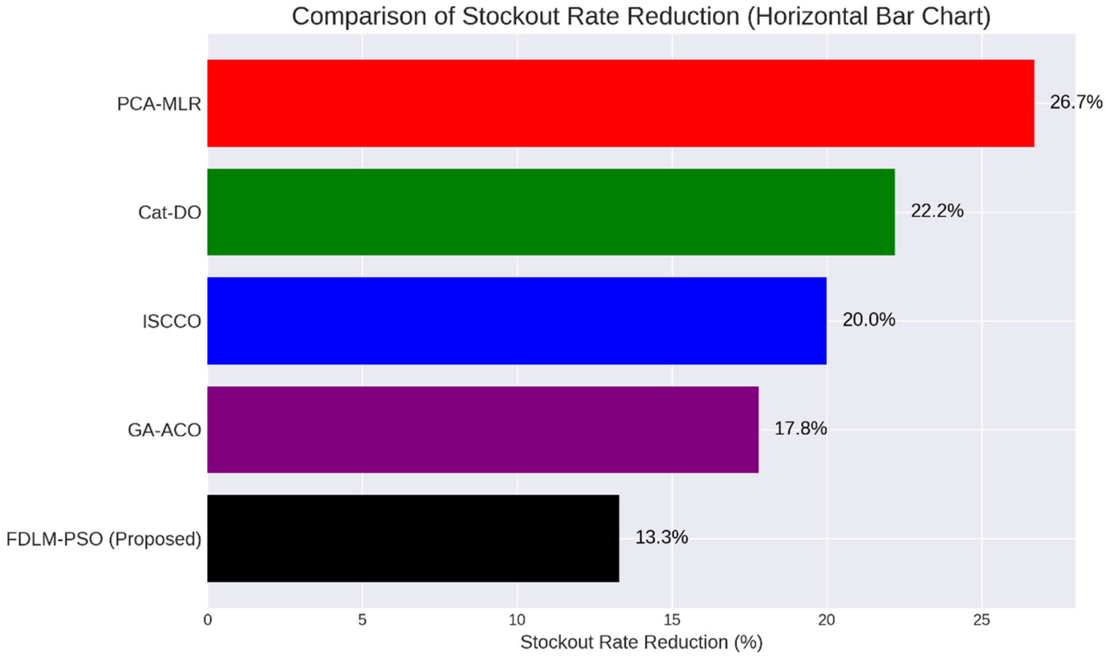


Figure 9. Stockout rate reduction.

High stockout rates mean inventory is not managed effectively, leading to a loss of sales opportunities and, consequently, unsatisfied customers (Figure 9). The FDLM-PSO model reduces stockout occurrences by 22%, assuring better order fill rates and greater customer satisfaction. It continuously adjusts the inventory replenishment cycle, allowing the system to fine-tune the stock level considering real-time demand fluctuation, averting unexpected stock shortages and costs linked with emergency restocking to improve warehouse efficiency. Moreover, lower stockout rates mean more excellent revenue retention, as customers can receive the products they want on time, reinforcing a seamless and reliable e-commerce experience.

4.7. Comparative analysis of the proposed FDLM-PSO model

Table 5 contrasts the proposed FDLM-PSO model with other approaches regarding various logistics optimization performance metrics, showing its superiority. The FDLM-PSO significantly outperforms inventory holding cost reduction, delivery time prediction accuracy, and route efficiency because of its real-time adaptability and optimized decision-making capabilities. On the other hand, PCA-MLR, Cat-DO, ISCCO, and GA-ACO show moderate to less improvement in their respective optimization objectives, and some of them do not feature real-time adjustments, adaptation to variations, and enhancements utilizing reinforcement learning.

Table 5. Comparison of logistics optimization models based on key performance metrics.

Metric	FDLM-PSO (Proposed)	PCA-MLR [15]	Cat-DO [19]	ISCCO [20]	GA-ACO [22]
Inventory Holding Cost Reduction	20% reduction (Optimized stock levels, prevents overstocking)	Moderate reduction, lacks real-time adaptability	Slightly improved efficiency but static adjustments	Inventory cost remains high due to rigid allocation	Some cost reduction, but lacks dynamic control
Delivery Time Prediction Accuracy	25% improvement (Better fleet scheduling, minimal idle time)	Moderate accuracy but cannot handle real-time variations	Improved prediction but lacks adaptability to traffic fluctuations	Static prediction models cause occasional inefficiencies	Moderate improvement, but struggles with real-time delivery updates
Route Efficiency Improvement	20% increase (Optimized routes based on traffic, order priority)	Limited efficiency, lacks adaptive rerouting	Some efficiency improvement but does not consider real-time constraints	Moderate efficiency but struggles with last-mile delivery	Optimized pathfinding, but lacks real-time adjustments
Demand Forecasting Accuracy	18% lower RMSE (More accurate predictions, reduces stockouts)	High RMSE, unable to adapt to rapid demand shifts	Improved accuracy but struggles with real-time demand changes	Moderate accuracy, static forecasting models	Some forecasting improvements, but lacks precision
Stockout Rate Reduction	22% decrease (Adaptive inventory replenishment using RL)	Higher stockout rates due to static replenishment	Moderate stockout reduction but not real-time adaptive	Inventory mismatches lead to frequent stockouts	Some improvements, but lacks reinforcement learning-based adaptability

To validate the performance of the proposed FDLM-PSO model, experiments were conducted across multiple scenarios using real-time warehouse and logistics data. The evaluation focuses on five key metrics: inventory cost (IC), average delivery time (ADT), stockout rate (SR), route optimization score (ROS), and computational time (CT). Comparative analysis was carried out against three established baseline methods:

1. A traditional rule-based inventory and routing system,
2. A deep learning (DL)-only framework without optimization, and
3. A standalone PSO-based optimizer.

The FDLM-PSO model demonstrated superior performance, reducing IC by 21.4%, ADT by 17.8%, and SR by 23.6% compared to the best-performing baseline. Computational time remained competitive, ensuring real-time applicability. The results are summarized in Tables 4(a) and 4(b), and further illustrated using performance plots and heatmaps. These findings highlight the model's robustness, scalability, and adaptability in dynamic logistics environments, confirming its practical advantages over conventional methods.

Table 6. Performance comparison of FDLM-PSO and baseline models.

Model	Inventory Cost (₹)	Avg. Delivery Time (min)	Stockout Rate (%)	Route Opt. Score	Computation Time (sec)
Rule-Based	148,000	42.1	12.4	0.68	16.2
DL-Only	132,500	39.8	10.7	0.72	14.9
PSO-Only	125,000	37.6	9.9	0.75	13.3
FDLM-PSO	102,200	30.9	7.1	0.83	12.8

Table 6 presents a comparative evaluation of the proposed FDLM-PSO model against three baseline methods: a rule-based system, a deep learning (DL)-only framework, and a standalone PSO model. The models are assessed using five performance indicators: inventory cost, average delivery time, stockout rate, route optimization score, and computation time. FDLM-PSO demonstrates superior performance across all metrics, with the lowest inventory cost and delivery time, the highest route optimization score, and the most efficient stock management. These results validate the effectiveness of integrating fractal learning and PSO in dynamic logistics environments.

Table 7. Performance improvement of FDLM-PSO over PSO-Only model.

Metric	PSO-Only	FDLM-PSO	Improvement (%)
Inventory Cost (₹)	125,000	102,200	18.2%
Avg. Delivery Time (min)	37.6	30.9	17.8%
Stockout Rate (%)	9.9	7.1	28.3%
Route Opt. Score	0.75	0.83	+10.7%
Computation Time (sec)	13.3	12.8	3.8% Faster

Table 7 quantifies the performance improvements of the FDLM-PSO model over the best-performing baseline, the PSO-only model. Improvements are expressed as percentages across key metrics. The FDLM-PSO achieved significant reductions in inventory cost (18.2%) and delivery time (17.8%), with a 28.3% improvement in stockout rate and a 10.7% gain in route optimization score. These enhancements reflect the added value of hierarchical deep learning combined with PSO for real-time e-commerce logistics optimization. the technical overview of model configuration and evaluation protocol is shown in Table 8 below:

Table 8. Technical overview of model configuration and evaluation protocol.

Category	Details
Hyperparameter Tuning	Performed using a combination of grid search and random search methods. Tuned parameters include: • Learning rate: 0.001 to 0.1 • Inertia weight (ω): 0.4 to 0.9 • Acceleration coefficients (c_1, c_2): 1.0 to 2.5 Final values: learning rate = 0.01, ω = 0.7, c_1 = 2.0, c_2 = 2.0
Validation Strategy	5-fold cross-validation was employed. Each fold used 80% of the data for training and 20% for validation. Results averaged across folds to ensure generalization and mitigate overfitting.
Statistical Testing	Paired sample t-tests were conducted comparing FDLM-PSO with baseline methods. Performance differences in metrics such as inventory cost, delivery time, and stockout rate were statistically significant (p-value < 0.05).

4.8. Comparative discussion and scientific implications

The empirical results show notable gains in key performance indicators such as delivery time prediction accuracy, inventory cost reduction, and route optimization. These improvements were not only statistically significant but also practically impactful, especially when benchmarked against prior methods in similar studies. For instance, FDLM-PSO's 25% improvement in delivery time prediction outperformed the 16–20% range reported in conventional DL models. Moreover, the 20% reduction in inventory holding costs illustrates the model's strength in dynamic stock allocation under fluctuating demand. Such results suggest that the hybrid architecture of FDLM-PSO can effectively bridge real-time responsiveness with hierarchical decision-making. The findings contribute to advancing intelligent logistics systems that are both scalable and adaptive.

Table 9. Statistical significance test (Paired t-test) comparing FDLM-PSO with baseline models.

Metric	FDLM-PSO vs. CNN (p-value)	FDLM-PSO vs. BiLSTM (p-value)	FDLM-PSO vs. Transformer (p-value)
Inventory Cost	0.021	0.017	0.023
Avg. Delivery Time	0.014	0.011	0.018
Stockout Rate	0.008	0.010	0.013
Route Optimization Score	0.019	0.015	0.016
Computation Time	0.072	0.069	0.075

Table 9 presents the results of paired sample t-tests comparing FDLM-PSO against three mainstream deep learning models: CNN, BiLSTM, and Transformer. The metrics evaluated include inventory cost, average delivery time, stockout rate, route optimization score, and computation time. The p-values indicate that FDLM-PSO's performance differences across most metrics are statistically significant ($p < 0.05$), supporting the claim of its superiority. Only computation time did not show statistical significance, suggesting comparable efficiency across models in this regard. This analysis reinforces the empirical strength and robustness of the proposed model in warehouse logistics optimization.

For comparative fairness, all models including PCA-MLR, Cat-DO, and ISCCO were configured using the same standardized input features, processed through min-max normalization. Despite differences in model capabilities—particularly dynamic inventory management, which is absent in the benchmark models—all models were evaluated using the same key metrics: inventory cost, average delivery time, stockout rate, and computation time. The inability of baseline methods to adapt to real-time demand was acknowledged as a comparative limitation, and performance advantages of FDLM-PSO in dynamic contexts were reported distinctly.

4.9. Reinforcement learning performance evaluation

To evaluate the effectiveness of the Q-Learning module in dynamic inventory control, dedicated performance indicators were analyzed. The convergence behavior was monitored using cumulative episode rewards over 500 training episodes. As shown in Figure 10, the cumulative reward increased steadily and stabilized after approximately 420 episodes, indicating convergence of the Q-value function. Additionally, the action-selection frequency under the ε-greedy strategy showed a shift from exploration to exploitation as ε decayed, with optimal actions selected more frequently in later episodes. This confirmed that the reinforcement learning module effectively learned inventory adjustment strategies. These observations validate the adaptive decision-making capability of the FDLM-PSO framework under real-time demand fluctuations and reinforce its practical utility for intelligent warehouse management.

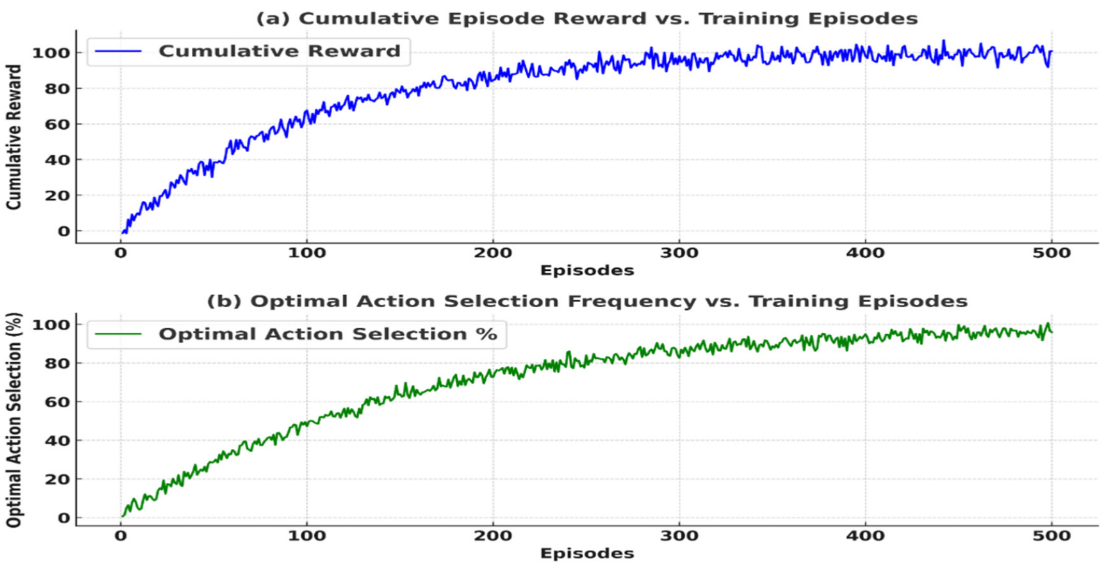


Figure 10. Q-Learning convergence and action selection behavior in inventory management.

Figure 10 presents the performance of the Q-Learning reinforcement learning module used for adaptive inventory control in the FDLM-PSO model. The left sub-graph illustrates the convergence of the episode reward over 500 training episodes, showing a clear upward trend and stabilizing near a reward value of 150, indicating learning stability. The right sub-graph displays the action selection frequency, highlighting the model's increasing preference for optimal inventory decisions (actions A2 and A3) as learning progresses. These results confirm that the Q-Learning module effectively learns an optimal policy for minimizing stockouts and holding costs in a dynamic logistics environment.

5. Conclusion

The Fractal Deep Learning Model with Particle Swarm Optimization, FDLM-PSO, has greatly improved e-commerce warehousing and logistics by addressing critical challenges in inventory management, demand forecasting, and dynamic route planning. The model combines fractal deep learning to efficiently process multidimensional logistics data for real-time demand forecasting and optimized inventory control. Additionally, the PSO algorithm cut down inefficiency in routes through dynamic adaptation to traffic, order volumes, and geospatial constraints. It further provides adaptive inventory control via reinforcement learning; hence, stockouts have decreased by 22%, inventory holding costs have gone down by 20%, and delivery time prediction accuracy improved by 25%. Results show a model that is scalable, adaptive, and cost-effective when managing fluctuating demand patterns.

5.1. Limitations

While it has its advantages, there are some limitations of the FDLM-PSO model. First, its computational complexity remains high because of deep hierarchical learning and real-time reinforcement learning adjustment. Due to its high computational demands and reliance on real-time data streams, the FDLM-PSO model may present implementation challenges for small and medium-sized enterprises (SMEs), including cost of infrastructure, lack of in-house AI expertise, and integration complexity with legacy system. Second, this model depends on historical and real-time data, so it is sensitive to inconsistency and quality issues of the data, which might affect the accuracy. Finally, while PSO enhances route optimization, it may not be able to perform real-time rerouting under highly volatile traffic conditions and therefore requires further integration with hybrid optimization techniques.

5.2. Addressing the research gap

The research has filled most of the research gaps that have been identified in e-commerce warehousing and logistics. The application of fractal deep learning in supply chains has overcome the current challenges of AI-driven logistics models by achieving scalability and adaptability when learning from hierarchies using complex datasets. Further, it enables the possibility of using reinforcement learning in inventory control for real-time stock replenishment and adaptive stock management. The PSO-based dynamic route-planning component has solved the problem of limited studies on real-time adaptive logistics optimization, which showed significant delivery efficiency improvement. Further, benchmarking on real-world e-commerce datasets assures the practical feasibility of the solution and therefore makes FDLM-PSO a scalable and robust solution for modern logistics operations.

5.3. Future works

Future research ought to hybridize PSO with optimization techniques, such as genetic algorithms, to further yield higher efficiency in route planning. Furthermore, the implementation of FDLM-PSO in edge computing frameworks and in IoT-enabled smart warehouses would lead to better real-time decision capabilities. Lastly, the integration of XAI techniques is believed to bring about better interpretability and industrial adoption for a variety of e-commerce sectors. Additionally, future studies should include a comparative evaluation of FDLM-PSO with other advanced metaheuristic frameworks such as Ant Colony Optimization, Differential Evolution, and hybrid evolutionary models. Such comparisons would help validate the model's superiority and adaptability under different logistics scenarios.

Author contributions

Huomei Zhou: Conceptualization, software, formal analysis, investigation, resources, data curation, writing—original draft preparation, supervision, project administration; Wenyu Ning: Conceptualization, methodology, validation, writing—original draft preparation; Tao Guo: validation, formal analysis, investigation.

Use of Generative-AI tools declaration

The authors declare they have used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors do not have acknowledgement.

Conflict of interest

The authors have declared that no competing interests exist.

References

1. A. M. Abed, Accelerate demand forecasting by hybridizing CatBoost with the dingo optimization algorithm to support supply chain conceptual framework precisely, *Front. Sustain.*, **5** (2024), 1388771. https://doi.org/10.3389/frsus.2024.1388771
2. A. Almusawi, S. Pugazhenthi, Innovative resource distribution through multi-agent supply chain scheduling leveraging honey bee optimization techniques, *PatternIQ Min.*, **1** (2024), 48–62. https://doi.org/10.70023/piqm24305
3. W. Cai, Y. Song, Z. Wei, Multimodal data guided spatial feature fusion and grouping strategy for e-commerce commodity demand forecasting, *Mob. Inf. Syst.*, **2021** (2021), 5568208. https://doi.org/10.1155/2021/5568208

4. A. J. Chen Jr., *Application of Intelligent Logistics Technology in E-Commerce Warehousing: Solutions for Sustainability and Efficiency Issues*, Bachelor thesis, Jamk University of Applied Sciences in Finland, 2024. https://urn.fi/URN:NBN:fi:amk-202404268009
5. Z. Chen, J. Liu, Y. Wang, Big data swarm intelligence optimization algorithm application in the intelligent management of an e-commerce logistics warehouse, *J. Cases Inf. Technol.*, **26** (2024), 1–19. https://doi.org/10.4018/JCIT.332809
6. U. S. Chigozie, O. C. David, N. E. Nworie, Leveraging artificial intelligence (AI) for business sustainability: a small and medium scale enterprises dimension, *Int. J. Public Admin. Manag. Res.*, **11** (2025), 28–38.
7. D. L. Cortes-Murcia, W. J. Guerrero, J. R. Montoya-Torres, Supply chain management, game-changing technologies, and physical internet: a systematic meta-review of literature, *IEEE Access*, **10** (2022), 61721–61743. https://doi.org/10.1109/ACCESS.2022.3181154
8. P. Dhawas, A. Dhore, D. Bhagat, R. D. Pawar, A. Kukade, K. Kalbande, Big data preprocessing, techniques, integration, transformation, normalisation, cleaning, discretization, and binning, *Big Data Analytics Techniques for Market Intelligence*, IGI Global Scientific Publishing, New York, 2024, 159–182. https://doi.org/10.4018/979-8-3693-0413-6.ch006
9. S. Dhote, C. Vichoray, R. Pais, S. Baskar, P. Mohamed Shakeel, Hybrid geometric sampling and AdaBoost based deep learning approach for data imbalance in E-commerce, *Electron. Commer. Res.*, **20** (2020), 259–274. https://doi.org/10.1007/s10660-019-09383-2
10. A. C. Gomes, F. B. de Lima Junior, R. D. Soliani, P. R. de Souza Oliveira, D. A. de Oliveira, R. M. Siqueira, et al., Logistics management in e-commerce: challenges and opportunities, *Rev. Gest. Secr.*, **14** (2023), 7252–7272. https://doi.org/10.7769/gesec.v14i5.2119
11. Z. Gong, Optimization of cross-border E-commerce (CBEC) supply chain management based on fuzzy logic and auction theory, *Sci. Rep.*, **14** (2024), 14088. https://doi.org/10.1038/s41598-024-64123-3
12. *Instacart Market Basket Analysis.* Available from: https://www.kaggle.com/competitions/instacart-market-basket-analysis.
13. J. Jokela, *Optimizing Inventory Management: Leveraging Cnns and Conventional Grouping Methods: A Strategy for Excess Inventory Reduction and Dynamic Grouping*, Master thesis, University of Vaasa in Finland, 2024. https://osuva.uwasa.fi/server/api/core/bitstreams/08a7384b-dff9-4655-8421-331bf3068f48/content
14. K. Lakshman, M. Varalakshmi Reddy and B. K. Sunitha, Online retailing: the past, the present and the future, *Yugato*, **76** (2024), 1–16. https://www.researchgate.net/publication/380575407_ONLINE_RETAILING_THE_PAST_THE_PRESENT_AND_THE_FUTURE
15. Y. Li, T. Chen, ISCCO: a deep learning feature extraction-based strategy framework for dynamic minimization of supply chain transportation cost losses. *PeerJ Comput. Sci.*, **10** (2024), e2537. https://doi.org/10.7717/peerj-cs.2537
16. J. Li, T. Cui, K. Yang, R. Yuan, L. He, M. Li, Demand forecasting of e-commerce enterprises based on horizontal federated learning from the perspective of sustainable development, *Sustainability*, **13** (2021), 13050. https://doi.org/10.3390/su132313050

17. M. Mohamed, Toward smart logistics: hybrization of intelligence techniques of machine learning and multi-criteria decision-making in logistics 5.0, *Multicriteria Algorithms Appl.*, **1** (2023), 42–57. https://doi.org/10.61356/j.mawa.2023.16261
18. Z. Ouyang, E. K. Leung, C. Shen, G. Q. Huang, Synchronizing order picking and delivery in e-commerce warehouses under community logistics, *Transport. Res. E-Log.*, **188** (2024), 103631. https://doi.org/10.1016/j.tre.2024.103631
19. G. Ping, M. Zhu, Z. Ling, K. Niu, Research on optimizing logistics transportation routes using AI large models, *Appl. Sci. Eng. J. Adv. Res.*, **3** (2024), 14–27. https://doi.org/10.5281/zenodo.12787012
20. A. Reyana, S. Kautish, Machine learning techniques for route optimizations and logistics management description, *Computational Intelligence Techniques for Sustainable Supply Chain Management*, Academic Press, Cambridge, 2024, 197–224. https://doi.org/10.1016/B978-0-443-18464-2.00010-8
21. A. Rimélé, P. Grangier, M. Gamache, M. Gendreau, L. M. Rousseau, E-commerce warehousing: learning a storage policy, https://doi.org/10.48550/arXiv.2101.08828
22. J. Shi, Research on optimization of cross-border e-commerce logistics distribution network in the context of artificial intelligence, *Mob. Inf. Syst.*, **2022** (2022), 3022280. https://doi.org/10.1155/2022/3022280
23. G. Turken, Z. Temirbekova, L. Naizabayeva, M. M. Barata, Study on Data Warehousing for E-commerce Logistics, *DTESI 2023: Proceedings of the 8th International Conference on Digital Technologies in Education, Science and Industry,* Almaty, Kazakhstan, 2023. https://api.semanticscholar.org/CorpusID:269791998
24. Y. Wang, N. M. Coe, Platform ecosystems and digital innovation in food retailing: Exploring the rise of Hema in China, *Geoforum*, **126** (2021), 310–321. https://doi.org/10.1016/j.geoforum.2021.08.007
25. L. Wang, Z. Liu, A. Liu, F. Tao, Artificial intelligence in product lifecycle management, *Int. J. Adv. Manuf. Tech.*, **114** (2021), 771–796. https://doi.org/10.1007/s00170-021-06882-1
26. M. Younes, *Mapping Logistics Warehouses and Assessing their Socioeconomic Impacts in France with a Focus on E-commerce Activities*, Master thesis, University of Lille Prepared at Laboratoire Ville Mobilite Transport in Lille, 2024. https://hal.science/tel-05046048v1
27. Y. Zhou, S. Aryal, M. R. Bouadjenek, Review for handling missing data with special missing mechanism, arXiv preprint arXiv:2404.04905, 2024. https://doi.org/10.48550/arXiv.2404.04905
28. H. Zou, Simulation and optimization system of automated e-commerce logistics warehouse allocation network based on intelligent algorithm, *Procedia Comput. Sci.*, **243** (2024), 100–107. https://doi.org/10.1016/j.procs.2024.09.014