A Hybrid ARIMA-LSTM Approach with Dual Optimization Modes for E-Commerce Warehouse Inventory-Sales Prediction and Allocation

Jianheng Lu, Shan He, Yurong Wu, Jiaxi Xing, Ning Wang, Yuan Long
School of Artificial Intelligence, Guangzhou Huashang College, Guangzhou, Guangdong, China

Abstract: This paper tackles e-commerce warehousing challenges through a data-driven approach. We preprocess interrupted data using interpolation techniques, then predict inventory via Auto ARIMA and sales via LSTM. Two optimization models are developed: a 0-1 integer programming model for cost-effective "one-product-one-warehouse" allocation, and a PSO algorithm for multi-objective "one-product-multiple-warehouses" planning. The proposed methods outperform baselines in cost reduction and operational efficiency, offering practical solutions for warehouse management

Keywords: Multi-Objective Optimization; Time Series Analysis; Auto-Regressive Integrated Moving Average Model; Long Short-Term Memory Network; Particle Swarm Optimization

1. Introduction

The rapid growth of global e-commerce, valued at $5.8 trillion in 2023 [1], has intensified competition in warehouse management efficiency. While previous studies have addressed inventory prediction [2] and warehouse optimization [3] separately, our work integrates both aspects through a novel hybrid approach. Unlike existing methods that focus solely on cost minimization [4], our model simultaneously considers five key objectives: capacity utilization, production efficiency, rental costs, category correlation, and warehouse allocation complexity.

2. Problem Re-statement

2.1 Problem Background

The commodity storage of e-commerce enterprises in each region is mainly undertaken by a group of warehouses. The stored commodities are divided and labeled according to attributes (categories, item types, etc.) for convenient inventory management. Due to the variety of commodity categories and large quantity of items, these commodities must be stored in different warehouses. Category warehousing planning determines which warehouses each commodity should be stored in. A reasonable category warehousing plan is crucial for improving the management efficiency of each warehouse and reducing the overall warehousing cost. Accurate warehousing freight volume prediction is an important basis for category warehousing planning. The accurate prediction results can predictably determine future warehousing resource use decisions, plan warehousing resources in advance, and reduce the investment in redundant sites.

Generally, this scenario requires predicting two targets, namely inventory and sales volume. The inventory refers to the total inventory of the category to be stored in all warehouses, and the warehousing result is limited by the warehouse capacity. The sales volume refers to the total quantity of the category to be packed and shipped out in all warehouses, and the warehousing result is limited by the production capacity. After obtaining the predicted freight volume of each category in the future, the category warehousing planning is an important research issue for supply chain planners. If the categories are concentrated in a small number of warehouses, it will exceed the capacity and production capacity limits of the warehouse, resulting in fulfillment problems. If the same category is distributed in multiple warehouses, it will significantly increase the number of warehouses, increasing the management difficulty and total cost of category inventory.

This scenario needs to consider two upper limits, namely the capacity upper limit and the production capacity upper limit. The capacity upper limit is the maximum inventory that a warehouse can store, and the production capacity upper limit is the maximum sales volume that a warehouse can ship out in a day. In addition, if similar categories (measured by category correlation) are placed in the same warehouse, the commodities in the same order are more likely to be shipped out centrally, which can reduce the number of packages in actual fulfillment and thus reduce the fulfillment cost.

2.2 Problem Description

Based on the above problem background, a model is established to solve the following specific problems:

Problem 1: Build a freight volume prediction model to predict the monthly inventory and daily sales volume of each category in the next three months. Accurate prediction of inventory and sales volume can support subsequent warehousing decisions, arrange warehousing resources in advance, and reduce redundancy and out-of-stock risks.

Problem 2: Assume that each category can only be placed in one warehouse in the "one-product-one-warehouse" warehousing scheme. Based on the prediction results of Problem 1, design an optimal warehousing scheme. It is necessary to find a balance among warehouse capacity, warehouse production capacity, and rent cost to achieve the lowest rent cost and efficient use of resources.

Problem 3: In the "one-product-multiple -warehouses" warehousing scheme, a category can be distributed in at most 3 warehouses, and categories with the same item type and high-level category should be placed in the same warehouse as much as possible. By optimizing category correlation, reduce order splitting and lower fulfillment costs. This is a multi-objective optimization problem that needs to balance category correlation, capacity and production capacity utilization, and rent cost.

3. Solution Ideas

First, data preprocessing was performed.

During the analysis of monthly inventory data across different product categories, significant data discontinuities were identified. To ensure the accuracy of subsequent analyses, linear interpolation was employed to address these gaps. Furthermore, for irregular missing values observed in the monthly sales data of various product categories, cubic interpolation was utilized for data imputation.

For the first question, the problem requires predicting and analyzing the inventory and sales volume of 150 categories in the next three months. Different models are used for these two predictions. For inventory, since the historical data lacks the data on the 1st of October-December 2022, linear interpolation can be used to fill the missing data points, and then the parameters of the model are determined based on the BIC matrix. The ARIMA algorithm is used to capture the linear characteristics of the time series and predict the inventory of different categories in the next three months. For sales data, due to the complex time-dependent relationship in the data and the large-scale missing data from October 2022 to March 2023, the first half of the dates are discarded for the data with too many interruptions. Then, cubic spline interpolation is carried out from April to June 2023, and then the prediction is carried out based on the LSTM recurrent neural network. Finally, the RMSE (root mean square error) is used to evaluate the performance of the model.

For the second question, the problem requires establishing a planning model to find the storage scheme for each category. In this question, some objective functions need to be established, including but not limited to: minimizing the total rent cost, maximizing the capacity utilization rate, maximizing the production capacity utilization rate, and maximizing the category correlation. And the question requires that each category can only be stored in one warehouse, so the constraint condition here is the one-product-one-warehouse limit. In addition, warehouse rent limit, warehouse capacity limit, and production capacity limit are set. The planning model can be solved using third-party libraries (such as PuLP, CVXPY). The basic solution process is: initialize the optimization problem, create decision variables, set the weights of each objective function, add the objective function, add constraint conditions, solve, and check the state of the solution.

For the third question, the problem requires establishing a planning model to find the warehousing scheme for each category. First, this question is a multi-objective optimization problem. An objective function is established to combine multiple factors into a single optimization objective through weighted summation. The factors to be considered include maximizing capacity utilization rate, maximizing production capacity utilization rate, minimizing total rent cost, minimizing the number of category warehousing, and maximizing category correlation. Second, according to the requirements of the question, constraint conditions are defined, including the capacity and production capacity upper limits of the warehouse, the limit of the number of warehouses for single-category storage, the correlation between categories, and the same item type and high-level category limit. Finally, the particle swarm optimization algorithm is used to apply PSO to this problem to find the optimal warehouse allocation scheme.

4. Methodology

4.1 Model Assumptions

4.1.1 Time series model (Auto ARIMA)

Assume that the data patterns in different time periods (2022 and 2023) are essentially consistent.

Assume that the data points on the 1st of each month are sufficient to represent the inventory status of the category in that month; assume that the daily data from July-September and April-June can effectively represent the sales pattern of the category, and this segmented data is sufficient to capture short-term sales rules.

Assume that the inventory data is already stationary or can be transformed into a stationary sequence through differencing and other means; assume that within these two time periods of daily sales data, external influencing factors (such as promotion, holiday influence, etc.) are relatively constant.

Assume that the parameter settings (autoregressive, differencing, moving average) of the ARIMA model can adapt to the trend of limited data.

4.1.2 Multi-objective optimization model (multi-objective-optimization)

(1) Warehouse rent cost assumption:

The rent cost of each warehouse does not change with the number or storage time of the stored categories, that is, as long as the warehouse is used, a fixed rent cost is generated.

The rent cost of the warehouse has nothing to do with the type or number of the stored categories, and only has to do with the activation state of the warehouse.

(2) Warehouse capacity limit assumption:

The capacity upper limit of each warehouse is fixed and does not change with time. Inventory adjustment needs to be completed within the capacity upper limit.

Assume that there is no situation of partial warehouse capacity locking, that is, all the capacity of the warehouse is available for any category, with high flexibility.

The volume of inventory occupation of different categories in the warehouse is the same (that is, the occupation of each category on the capacity can be directly accumulated through the quantity), which can simplify the capacity calculation.

(3) Warehouse production capacity limit assumption:

The outbound capacity of each warehouse is fixed, that is, the production capacity upper limit remains unchanged throughout the optimization cycle.

The outbound operations of various categories in a single warehouse do not interfere with each other, that is, the outbound of multiple categories can be carried out independently.

Assume that the outbound capacity is not limited by the inventory. The outbound production capacity upper limit of the warehouse is the upper limit of the total outbound volume of all categories.

(4) Inventory and outbound operation assumption:

Assume that inventory adjustment and outbound operations can be arranged arbitrarily within the time period of the model, without being restricted by the actual operation duration or scheduling.

The inventory of each category can be flexibly allocated to different warehouses, without considering factors such as transfer time or inventory transfer time.

(5) Category correlation assumption:

Assume that the correlation between categories does not change with time and is known and stable within the optimization cycle.

The correlation between categories can be independent of inventory or sales volume, that is, as long as the categories are stored in the same warehouse, correlation will be generated, regardless of the quantity relationship.

(6) Time cycle and operation synchronization assumption:

Assume that the operations of all warehouses are carried out synchronously, that is, inventory allocation, outbound, and rent calculation are all completed within the same time cycle for unified optimization.

The lease, inventory, and outbound of the warehouse are re-evaluated within this fixed cycle, without considering the continuity problem across cycles.

(7) Cost and efficiency linear relationship assumption:

Assume that the optimization of total rent cost, capacity utilization rate, production capacity utilization rate, and category correlation is linear, so that linear programming tools (such as PuLP and cvxpy) can solve more efficiently.

The mutual influence between each objective function can be calculated independently in the model to avoid non-linear problems of cross-influence.

4.2 Symbol Explanation

For the convenience of the establishment and solution process of the following models, the following explanations are given for the key symbols used, as shown in Table 1.

Table 1. Symbol Definitions and Descriptions

Symbol	Description
x, y	Data points
a_i, b_i, c_i, d_i	Equation coefficients
$S_i(x)$	Polynomial
p	Autoregressive order
q	Moving-average order
d	Order of differencing
B	Lag operator
N	Number of data points
w	Weight
c_1, c_2	Learning factors

4.3 Data Cleaning

4.3.1 Data preprocessing

To facilitate subsequent analysis, the dates in the inventory data and sales data were standardized into a unified datetime format and sorted by product category and time.

In the inventory dataset, data gaps and discontinuities were observed from October to December 2022. To enhance the accuracy of analytical results, linear interpolation was applied to generate estimated values on the 1st day of each missing month.

For the sales data, due to significant irregularities, only the period from April to June 2023-which exhibited relatively complete records-was selected for analysis.

4.3.2 Data filling based on linear interpolation algorithm

Principle and Function of Linear Interpolation:

(1) Principle:

1) Basic concept: Linear interpolation assumes that the data changes linearly between two known data points. Given two points (x_0, y_0) and (x_1, y_1), linear interpolation can connect these two points with a straight-line segment.

2) Interpolation formula: For a point x to be estimated (where $x_0 < x < x_1$), the calculation formula of linear interpolation is:

$$y = y_0 + \frac{(y_1 - y_0)}{(x_1 - x_0)} \cdot (x - x_0) \qquad (1)$$

This formula means that the value of y is estimated according to the position of x relative to x_0 and x_1, through a proportional relationship.

(2) Function:

1) Data estimation: In data analysis and scientific research, linear interpolation is often used to fill missing data and provide estimated values within the known data range.

2) Data smoothing: By inserting new values between data points, linear interpolation can help smooth the data set, making the data show a more continuous trend.

3) Graph drawing: In graphic calculation and computer graphics, linear interpolation is used to draw smooth curves and paths, especially in animation and game development.

4) Simplicity and computational efficiency: Linear interpolation is relatively simple and has high computational efficiency, suitable for processing small data sets and real-time applications.

(3) Linear Interpolation of Inventory Data:

When observing the inventory data of 350 categories, it is found that it only contains the inventory on the 1st of each month from July to September 2022 and from January to June 2023 for each category, that is, less than one-year data volume. If the time-series model is directly trained with this data, the prediction effect will obviously be poor. Therefore, by nesting the interpolate function within a loop, we achieved the "continuity" of inventory data for the 1st of each month from July 2022 to June 2023 across all categories. Specifically, this approach filled in the missing inventory data for the 1st of October, November, and December 2022. Consequently, using this processed data to train the model is expected to yield significantly improved prediction performance compared to the initial dataset.

In general, linear interpolation is a basic and effective interpolation method, suitable for various practical applications, especially when data is missing or the model needs to be simplified. Although it assumes that the data changes linearly, in many cases, linear interpolation can still provide sufficiently accurate estimation results.

4.3.3 Data filling based on cubic spline interpolation algorithm

During preliminary analysis of the sales data, we identified severe data discontinuities and inconsistencies in the period from October 2022 to March 2023. Given these substantial data quality issues, this earlier timeframe was excluded from our study to ensure the robustness of analytical results. Consequently, our primary analysis focuses on the more complete and reliable dataset spanning April to June 2023.

Cubic spline interpolation is a commonly used numerical method for smooth interpolation between known data points. It constructs a series of cubic polynomials to achieve interpolation, so that the function not only has the same function value at each data point, but also the first-order and second-order derivatives are continuous.

Suppose we have a set of data points $(x_0, y_0), (x_1, y_1)\ldots(x_n, y_n)$, where y_i is the known value at x_i. The goal is to find a cubic polynomial $S_i(x)$ on each interval $[x_i, x_{i+1}]$ such that:

$$S_i(x_i) = y_i \quad (2)$$

$$S_i(x_{i+1}) = y_{i+1} \quad (3)$$

$$S_i^{'}(x_{i+1}) = S_{i+1}^{'}(x_{i+1}) \quad (4)$$

$$S_i^{''}(x_{i+1}) = S_{i+1}^{''}(x_{i+1}) \quad (5)$$

On each interval $[x_i, x_{i+1}]$, the cubic spline polynomial $S_i(x)$has the form:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (6)$$

At each node, the value of the spline function is equal to the known value:

$$S_i(x_i) = y_i \quad (7)$$

$$S_i(x_{i+1}) = y_{i+1} \quad (8)$$

At each internal node, the first-order and second-order derivatives of the spline function are continuous:

$$S_i^{'}(x_{i+1}) = S_{i+1}^{'}(x_{i+1}) \quad (9)$$

$$S_i^{''}(x_{i+1}) = S_{i+1}^{''}(x_{i+1}) \quad (10)$$

Common boundary conditions include natural boundary conditions (the second derivative is zero at the boundary) or clamped boundary conditions (the first derivative is specified at the boundary).

Through the above conditions, a linear equation system can be obtained to solve the coefficients a_i, b_i, c_i, d_i. The specific steps are as follows:

1) Set $h_i = x_{i+1} - x_i$ as the interval length.

2) Through the continuity condition, an equation about c_i is obtained:

$$c_{i+1} = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}) \quad (11)$$

3)Using the second-derivative continuity condition, a system of equations is constructed:

$$\frac{c_{i+1} - c_i}{h_i} = \frac{c_i - c_{i-1}}{h_{i-1}} \quad (12)$$

4) The boundary conditions can be set as natural boundary conditions:

$$c_0 = 0, c_n = 0 \quad (13)$$

5) Solve the system of equations to obtain all coefficients $c1$, and then use the interpolation conditions and first-derivative continuity to solve a_i, b_i, d_i.

6) Since the cubic spline interpolation for sales volume cannot be less than 0, and to ensure that the data is not distorted, on the basis of the original algorithm, the interpolated value is restricted to be not less than 0 and not greater than the existing value. At the same time, the interpolation result is rounded to an integer.

7) Finally, by solving the above linear equation system, the polynomial coefficients on each interval are obtained, and thus a complete cubic spline interpolation function is constructed.

8) After interpolation, we can check whether the dates of each category are continuous.

After interpolation, we verified the temporal continuity of the data for each category. The results demonstrate that the cubic spline interpolation method employed in this study effectively fills the missing data while adhering to two critical constraints: (1) the interpolated values remain non-negative, and (2) they do not exceed the maximum observed value in the existing dataset. Moreover, the interpolation preserves the original data's trend and volatility characteristics, ensuring a smooth and logically consistent transition between observed and interpolated values. This approach robustly maintains the dataset's integrity and coherence for subsequent analysis.

4.4 Model Construction and Solution

4.4.1 Problem 1 model establishment and solution

(1) Predicting inventory in the next three months based on the ARIMA algorithm

First, an ARIMA model is established.

The essence of the ARIMA model is a combination of differencing operations and the ARMA model, denoted as ARIMA (p, d, q). The ARIMA model can be expressed as:

$$\phi(B)(1-B)^d y_t = \theta(B)\varepsilon_t \quad (14)$$

where y_t is the time series of historical observed values, d is the order of differencing, p and q are the order of the autoregressive model and the moving average of past observations respectively, ε_t is an independent and identically distributed white-noise sequence with a mean of zero and a constant variance. B is the lag operator, and B satisfies the following expression:

$$B^n y_t = y_{t-n} \quad (15)$$

$$\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p \quad (16)$$

$$\theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q \quad (17)$$

The key to establishing an ARIMA (p, d, q) model lies in the selection of the three parameters (p, d, q) [5]. d is the order of differencing, and the purpose of differencing is to transform the original observed value sequence into a stationary time series. In this paper, the Bayesian Information Criterion (BIC) is used to select p and q. The Bayesian information criterion can give a simple approximate logarithmic model evidence, as shown below:

$$BIC = Accuracy(m) - \frac{p}{2} logN \quad (18)$$

Where p is the number of parameters, and N is the number of data points.

However, when actually predicting the inventory of 350 categories in the next three months, the Auto ARIMA model is used instead of the ARIMA model. The reason is that although the ARIMA model is a very powerful model for predicting time-series data, its data preparation and parameter adjustment process is very time-consuming. If the ARIMA model is used to predict the inventory of 350 categories, in order to ensure more accurate prediction results, different p, d, q parameters should be set for different categories. As a result, such a long preparation process makes the entire prediction process extremely inefficient. A methodology that sacrifices efficiency too much for a certain accuracy is definitely not advisable. Therefore, the Auto ARIMA model is adopted, which makes the entire task more efficient while ensuring a certain accuracy.

(2) Sales volume prediction based on the LSTM long-short-term memory recurrent neural network

First, introduce LSTM.

The memory persistence of the human brain enables it to use past experiences to understand new information. Traditional neural networks, such as feed-forward neural networks, lack a memory mechanism and cannot use the learning results of previous layers for subsequent inference learning. In order to simulate this memory function of the human brain, the recurrent neural network (RNN) is proposed. It allows the network to consider past information when processing the current input through cyclic connections.

However, the RNN encounters the long-term dependency problem when processing long-sequence data, that is, it is difficult to capture the long-distance dependency relationship in the time sequence. To solve this challenge, the LSTM (Long-Short-Term Memory Network) is introduced. It controls the flow of information by introducing a gating mechanism, effectively solving the problems of long-term dependency and gradient vanishing/explosion [6], enabling the network to learn important patterns in long-sequence data. LSTM is widely used in many fields such as natural language processing and speech recognition due to its advantages in processing time-series data.

Next, use LSTM to predict the sales data [6].

First, extract the sales data and normalize it. Here, the Min Max Scaler in sklearn is used for processing. Second, split the data into input sequences and target value sequences, and reshape the input sequences into the shape required by the LSTM model, that is, a three-dimensional vector. Move the data with a step size of 1 to obtain several input sequences with 10 items in each group, and the 1 item after 10 items is the target value. Follow the movement to obtain the target value sequence. Third, build the model. Add several LSTM layers to the Sequential container and use Dense as the output layer to build a neural network. Fourth, compile the model, select an optimizer and a loss function. Fifth, train the model to predict the sales data every 10 days from July to September 2023. In this process, the input sequence will be continuously updated and new predicted values will be added to ensure that the data is continuously moving. Sixth, perform denormalization on the sales data predicted by the model.

4.4.2 Problem 2 model establishment and solution

(1) Solving the selection problem based on 0-1 integer programming

0-1 integer programming is a special type of integer programming problem [7] where decision variables can only take values of 0 or 1. It is often used to solve selection problems, such as whether to choose a certain project or whether to execute a certain task. Decision variables usually represent "selection" (x_i=1) or "non-selection" (x_i=0).

(2) Finding the optimal solution of the model based on the pulp library

First, introduce PuLP.

PuLP is a Python library for linear programming (LP), integer linear programming (ILP), and mixed-integer linear programming (MILP) problems. The full name of PuLP is "Python for Mathematical Programming". It provides a simple and powerful tool that allows users to define optimization problems, build mathematical models, and use different solvers to solve them.

One of the main features of PuLP is its ease of use. It allows users to define optimization problems in a simple way without in-depth knowledge of the details of mathematical programming. The syntax design of PuLP is aimed at enabling users to intuitively express the constraints and objective functions of the problem. This concise and clear syntax makes PuLP an ideal choice for solving linear programming problems, especially for users who are not very familiar with the field of mathematical programming. In PuLP, users can easily define variables, constraints, and objective functions. Through simple API calls, users can specify the upper and lower bounds of variables, the coefficients of constraint conditions, and the coefficients of the objective function. PuLP also provides checks and displays of problem characteristics to help users verify the correctness of the model. The use of PuLP is not limited to linear programming problems, but can also handle integer linear programming and mixed-integer linear programming.

Apply PuLP to determine the optimal allocation strategy for product categories across warehouses.

The basic idea of building the model is as follows: First, initialize the optimization problem, that is, determine that the goal is to minimize the objective function. Second, create decision variables, that is, set a 350-row and 140-column two-dimensional binary variable matrix and a one-dimensional auxiliary binary variable matrix. Among them, the 350 rows of the two-dimensional binary variable matrix represent 350 categories, and the 140 columns represent 140 warehouses. A solution of 0 means that the corresponding category is not stored in the corresponding warehouse, and a solution of 1 means that the corresponding category is stored in the corresponding warehouse. The variables in the one-dimensional auxiliary binary variable matrix represent whether 140 warehouses are used. Third, add each objective function and combine them into a final objective function using the weight method, as shown below:

Objective function 1: Capacity utilization rate (f_1)

$$max f_1 = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{x_{i,j} \cdot \hat{P}_i}{C_j^{max}} \quad (19)$$

where $\hat{P}_i$ represents the predicted average monthly inventory of category i, and C_j^{max} represents the capacity upper limit of warehouse j.

Objective function 2: Production capacity utilization rate (f_2)

$$max f_2 = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{x_{i,j} \cdot \hat{E}_i}{W_j^{max}} \quad (20)$$

where $\hat{E}_i$ represents the predicted daily sales volume of category i, and W_j^{max} represents the production capacity upper limit of warehouse j.

Objective function 3: Total rent cost (f_3)

$$min f_3 = \sum_{j=1}^{N} T \cdot r_j \cdot \sum_{j=1}^{N} x_{i,j} \quad (21)$$

where T represents the total number of days in the planning cycle, and r_j represents the daily cost of warehouse j.

Objective function 4: Category correlation (f_4)

$$maxf_4 = \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k=1}^{M}\frac{g_{i,k}\cdot x_{i,j}\cdot x_{k,j}}{2} \quad (22)$$

where $g_{i,k}$ is the correlation coefficient between category i and category k.

The total rent cost (minimization): remains unchanged.

The capacity utilization rate (maximization): is transformed into a minimization problem by taking the negative value.

The production capacity utilization rate (maximization): is transformed into a minimization problem by taking the negative value.

The category correlation (maximization): is transformed into a minimization problem by taking the negative value.

The combined objective function is as follows:

$$-w_1\cdot f_1 - w_2\cdot f_2 + w_3\cdot f_3 - w_4\cdot f_4 \quad (23)$$

Fourth, add each constraint condition, as shown below:

Constraint 1: The total inventory of a single warehouse does not exceed the capacity upper limit (total inventory of a single warehouse/capacity upper limit)

$$\sum_{i=1}^{M} x_{i,j}\cdot P_i \le C_j^{max} \quad (24)$$

$$\sum_{j=1}^{N} x_{i,j}\cdot P_i = P_i^{max} \quad (25)$$

Constraint 2: The total outbound volume of a single warehouse does not exceed the production capacity upper limit (total outbound volume of a single warehouse/production capacity upper limit)

$$\sum_{i=1}^{M} x_{i,j}\cdot E_i \le W_j^{max} \quad (26)$$

$$x_{i,j}\cdot E_i \le x_{i,j}\cdot P_i \quad (27)$$

$$\sum_{j=1}^{N} x_{i,j}\cdot E_i = E_i^{max} \quad (28)$$

where the quantity of category i cannot exceed the production capacity upper limit of the warehouse, the production capacity of category i in warehouse j cannot exceed its inventory in this warehouse, and the total production capacity of category i is the predicted value of sales volume.

Constraint 3: The number of warehouses for single-category storage is limited to 1

$$\sum_{j=1}^{N} x_{i,j} = 1 \quad (29)$$

Constraint 4: Ensure that if the warehouse is used, the auxiliary variable $y[j]$ is 1

$$\sum_{j=1}^{N} x_{i,j} \le 350 * y_j \quad (30)$$

Fifth, solve and view the state of the solution.

4.4.3 Problem 3 definition and model construction

(1) Determining the optimization objectives

Building upon Problem 2, a supplementary objective function f_5 is incorporated into the model.

Objective functions:

Capacity utilization rate (f_1)

Production capacity utilization rate (f_2)

Total rent cost (f_3)

Category correlation (f_4)

Number of category warehousing (f_5):

$$minf_5 = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{i,j} \quad (31)$$

Combined objective function：

Using the weighted summation method, by assigning a weight to each objective[8], multiple objectives are combined into a single objective function. A single objective function is used to approximate the multi-objective optimization problem. Since there are both maximization and minimization objectives in this problem, each objective needs to be converted to the same direction through positive and negative conversion. Considering comprehensive factors, all the multi-objectives in this problem are converted into minimization problems. Among them, achieving the maximization objective means finding the solution with the largest negative value after conversion, and achieving the minimization objective does not need to be converted, that is, finding the minimum value. Through conversion, it is convenient to find the optimal value and update the particles later. The conversion process is as follows:

The capacity utilization rate (maximization): is converted into a minimization problem by taking the negative value.

The production capacity utilization rate (maximization): is converted into a minimization problem by taking the negative value.

The total rent cost (minimization): remains unchanged.

The category correlation (maximization): is converted into a minimization problem by taking the negative value.

The number of category warehousing (minimization): remains unchanged.

The combined objective function is as follows:

$$-w_1\cdot f_1 - w_2\cdot f_2 + w_3\cdot f_3 - w_4\cdot f_4 + w_5\cdot f_5 \quad (32)$$

(2) Determining decision variables and

constraint conditions

Create decision variables based on the zero-one planning problem:

Decision variables: $x_{i,j}$, a vector with a length of M*N, represents the state of each category in each warehouse. It is a binary decision variable. This means that each $x_{i,j}$ can only take one of two values:

$x_{i,j}$=0: Represents the scenario where the i-th product category is excluded from the j-th warehouse.

$x_{i,j}$=1: Represents the scenario where the i-th product category is allocated to the j-th warehouse.

$$\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \quad (33)$$

Category-warehouse matrix $x_{i,j}$

Determining constraint conditions:

According to the requirements of the question and observing the data set, there are multiple constraints:

Constraint 1: The total inventory of a single warehouse does not exceed the capacity upper limit (total inventory of a single warehouse/capacity upper limit)

Constraint 2: The total outbound volume of a single warehouse does not exceed the production capacity upper limit (total outbound volume of a single warehouse/production capacity upper limit)

Constraint 3: The number of warehouses for single-category storage is limited between 1 and 3

$$1 \leq \sum_{j=1}^{N} x_{i,j} \leq 3 \quad (34)$$

Constraint 4: Categories with the same high-level category and the same item type are stored in the same warehouse

$$k_i^{g,q} = k_k^{g,q}, (x_{i,j}, x_{k,j} = 1, i, k \in (1, N)) \quad (35)$$

where $k_i^{g,q}$ represents the high-level category and item type of category i. The $k_i^{g,q}$ of categories existing in warehouse j should be equal.

(3) Establishing and solving the model based on the particle swarm algorithm

The particle swarm algorithm (PSO) [9] belongs to a type of swarm intelligence algorithm and is designed by simulating the foraging behavior of bird flocks.

Suppose there is only one piece of food (that is, the optimal solution in the commonly mentioned optimization problem) in the area. The task of the bird flock is to find this food source. During the entire search process, the bird flock transmits each other's information, allowing other birds to know their positions. Through such cooperation, they judge whether the solution they have found is the optimal solution, and at the same time, transmit the information of the optimal solution to the entire bird flock. Finally, the entire bird flock can gather around the food source, that is, the optimal solution is found, and the problem converges [10].

The defined parameters are listed in Table 2.

Table 2. Details of PSO-defined Parameters

Parameter	Definition
M	Number of product categories
N	Number of warehouses
P_i	Predicted inventory level
E_i	Predicted sales volume
r_j	Daily rental cost
$g_{i,k}$	Correlation coefficient
C_j^{max}	Maximum storage capacity
W_j^{max}	Maximum throughput capacity
T	Planning period
$k_i^{g,q}$	Advanced category & pack
w_1- w_5	Weights

Define the particle class:

Particle attributes:

Position (position): Represents the current solution

Velocity (velocity): Guides the movement of the particle in the solution space

Best position (best_position): The best solution found by the particle in history

Particle methods:

Update_velocity: Updates the velocity according to the individual best position and the global best position

Update_position: Updates the position according to the velocity and ensures that the new position satisfies the constraints

Update_best: Updates the optimal position and optimal value of the particle.

Implement the PSO algorithm:

The algorithmic procedure is outlined below:

Particle representation: Each particle represents a potential solution, that is, the allocation ratio of each category in each warehouse.

Particle swarm initialization: Create a certain number of particles.

Iterative optimization:

For each particle:

Update the velocity and position: Update the velocity and position of the particle according to the individual best position and the global best position.

The standard formula for updating the velocity is as follows:

$$v_{id} = w \cdot v_{id} + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (p_{gd} - x_{id}) \quad (36)$$

where v_{id} is the velocity of particle i in dimension d.

w is the inertia weight, which controls the influence of the previous value of the particle velocity on the current velocity.

c_1 and c_2 are acceleration coefficients, also known as learning factors, which control the tendency of the particle to move towards the individual best position and the social best position.

r_1 and r_2 are two random numbers, usually between 0 and 1, which introduce randomness to the algorithm.

p_{id} is the historical best position of particle i in dimension d.

x_{id} is the current position of particle i in dimension d.

p_{gd} is the coordinate of the global best position in dimension d.

Check constraints: Ensure that the new position satisfies all constraint conditions, that is, ensure that the total inventory of a single warehouse does not exceed the capacity upper limit;

ensure that the total outbound volume of a single warehouse does not exceed the production capacity upper limit;

ensure that the distribution ratio of the same category does not exceed 3 warehouses;

ensure that categories with the same high-level category and the same item type are placed in the same warehouse as much as possible.

Calculate the objective function value: Calculate the objective function value of the current position.

Update the individual and global optima: If the objective function value of the current position is better than the individual historical best value, update the individual best position; if it is better than the global best value, update the global best position.

Termination condition: Reach the maximum number of iterations or the objective function value converges.

(4) Solving process

Algorithm parameter settings

Number of particles: 30

Number of iterations: 100

Execution of the PSO Algorithm

The PSO algorithm is executed by invoking the PSO function, passing the objective function, boundary conditions, data, and other parameters as inputs.

Output results

The optimal warehouse allocation scheme (best_position) and the corresponding objective function value (best_value) are printed.

5. Experimental Result

5.1 Inventory Forecasting Results Using ARIMA

Based on the Auto ARIMA model, the inventory of each category is predicted. Some of the results are shown in Table 3 below.

Table 3. Monthly Stock Prediction Results

	Jul Stock	Aug Stock	Sep Stock
category1	6177	6177	6177
category31	19	1	0
category61	235566	235566	235566
category91	5864	5117	4369
category121	75154	75154	75154
category151	124814	124814	124814
category181	715	643	596
category211	7863	8516	9349
category241	61870	64974	60062
category271	14793	14793	14793
category301	2781	2781	2781
category331	58075	62911	67747

5.2 Monthly Sales Prediction with LSTM Networks

Based on the LSTM model, the monthly sales volume of each category is predicted.

The sales prediction results demonstrate significant variations across product categories. Category 61 remains the top performer with over 1,700 units sold, though showing a slight decline of 14 units (from 1,717 to 1,703) over 30 days. Category 151 follows an inverted-U pattern, reaching its peak of 1,613 units in mid-August before decreasing by 4.4%. Steady growth is observed in Category 301 with a 15.6% increase (from 32 to 37 units) within 35 days, while Categories 211 and 271 exhibit more moderate growth rates of 12.5% and 2.3% respectively. Three categories (1, 91 and

181) maintain exceptional stability with minimal fluctuations within ±1 unit. Notably, Category 31 continues to record zero sales throughout the period.

5.3 Optimal Allocation via Binary Integer Programming

Based on a 0-1 integer programming model, the optimal warehousing scheme is obtained. Some of the results are shown in Table 4 below.

5.4 Multi-Warehouse Allocation Using MOPSO

Based on a multi-objective particle swarm optimization (MOPSO) model, the optimal allocation strategy is derived, and partial results are presented in Table 5.

Table 4. "One-product-one-warehouse" Allocation Results

	warehouse
category1	123
category31	140
category61	68
category91	76
category121	92
category151	26
category181	80
category211	61
category241	103
category271	110
category301	60
category331	101

Table 5. Daily Sales Forecast Results

	Warehouse	Warehouse	Warehouse
category1	101		
category31	6		
category61	64	92	93
category91	101		
category121	63	90	91
category151	25	76	110
category181	66	124	138
category211	101		
category241	18	117	120
category271	2	3	4
category301	96	97	140
category331	97	98	140

6. Conclusion

6.1 Model Advantages

This study conducts a comprehensive analysis of the predicted inventory levels and sales volumes of product categories, covering various aspects such as model establishment, data preprocessing, and comparative analysis of multiple algorithms. Linear interpolation and cubic spline interpolation are utilized to refine the data, ensuring a thorough resolution of the problem.

Multiple models are employed in this study, making full use of the dataset and ensuring a complete data processing procedure.

A combination of various mathematical models, including ARIMA and LSTM, is applied, and their superior performance in predicting inventory levels and sales volumes is validated through comparative analysis.

The study leverages the PuLP algorithm and Particle Swarm Optimization (PSO) to optimize the objective function, enhancing the efficiency and quality of the solution. This provides timely information support for inventory management and sales strategies.

6.2 Model Limitations

Due to data interruptions and irregular missing values, the accuracy and stability of the model are limited when processing such data. This poses challenges in determining model parameters using the ARIMA algorithm.

For sales data, the LSTM model’s predictive performance may not meet expectations due to complex temporal dependencies and a significant amount of missing data.

For large-scale problems, PuLP may not be as fast as specialized commercial solvers like CPLEX or Gurobi. Its performance may degrade when handling large or complex linear programming problems.

6.3 Model Extensions

To address data interruptions and missing values, more advanced data interpolation and imputation techniques, such as higher-order interpolation methods, can be explored to improve the quality of data preprocessing.

The model can be applied to other similar time series prediction problems, such as financial market analysis and weather forecasting, to validate its universality and effectiveness.

Given the constraints of time and data points, further optimization of ARIMA model parameters and the PuLP library can be pursued.

Acknowledgments

This research was supported by the project

"Research on Deep Learning-Based Object Detection Algorithms" (No. 2022HSXS087).

References

[1] Statista Digital Market Outlook. Global E-commerce Market Size 2023. Statista GmbH. Retrieved from
[2] Y. Zhang, Research on inventory optimization of L cross-border e-commerce company's overseas warehouses based on inventory classification and demand forecasting, Master's thesis, Zhejiang Univ., Hangzhou, China, 2024.
[3] J. Jiang, J. Han, and Y. Zhang, Research on optimization of storage location assignment in e-commerce warehouses, Mechanical Design and Manufacturing, 2025, 1-6.
[4] Li J, Liu R, Zhang X, et al. Constrained multi-objective evolutionary algorithm based on the correlation between objectives and constraints. Swarm and Evolutionary Computation, 2025, 94101903-101903.
[5] Hyndman, R. J., & Athanasopoulos, G. Forecasting: Principles and Practice (3rd ed.). OTexts. 2021.
[6] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks, Int. J. Forecast. 2020, 36(3): 1181-1191.
[7] S. Mayadunne, H. K. Rajagopalan, and E. Sharer, A multi-step mixed integer programming heuristic for warehouse layout optimization, Supply Chain Analytics, 2024, 8, p. 100088.
[8] R. T. Marler and J. S. Arora, The weighted sum method for multi-objective optimization: New insights, Struct. Multidiscip. Optim, 2010, 41(6): 853-862.
[9] Z. Tan, H. Li, and X. He, Optimizing parcel sorting process of vertical sorting system in e-commerce warehouse, Advanced Engineering Informatics, 2021, 48, 101279.
[10]B. L. Shankar, S. Basavarajappa, J. C. H. Chen, and R. S. Kadadevaramath, Location and allocation decisions for multi-echelon supply chain network-A multi-objective evolutionary approach, Expert Syst. Appl, 2013, 40(2): 551-562.

