RAIRO-Oper. Res. 60 (2026) 2227–2252
https://doi.org/10.1051/ro/2026066

RAIRO Operations Research
www.rairo-ro.org

RESEARCH ON SUPPLY CHAIN MANAGEMENT AND ORDERING STRATEGIES FOR PERISHABLE GOODS IN FARMERS' MARKETS

Yuchi Li[1,⋆], Zhanli Li[2,⋆], Lili Xia[1,*], Jiqing Sun[1], Jianhao Li[1] and Yuting Xu[1]

Abstract. Given the uncertainty in market demand and the complexity of supply chain management for perishable vegetable products, traditional demand forecasting and pricing strategies face substantial challenges in practical applications. To address these issues, this study proposes a KAN-LSTM-based framework for demand forecasting and ordering optimization. By integrating the nonlinear representation capability of KAN with the long-term dependency modeling ability of LSTM, the proposed framework provides an effective approach for forecasting perishable vegetable sales and procurement prices. In addition, a multivariate stepwise regression model is employed to estimate price elasticity and support pricing decisions. The ordering strategy is then optimized using the McCormick envelope method combined with the SLSQP algorithm. Experimental results show that the KAN-LSTM model achieves high forecasting accuracy and stable performance in both sales and price prediction tasks. Compared with xLSTM, TCN, and Transformer models, KAN-LSTM demonstrates favorable predictive performance for the dataset considered in this study. The optimization results further indicate that the proposed framework is effective in improving profit and cost control, highlighting its practical value for perishable goods supply chain management in farmers' markets.

Mathematics Subject Classification. 68T05, 65K05, 90B06.

Received April 12, 2025. Accepted May 27, 2026.

1. Introduction

Demand forecasting and price optimization are pivotal components of modern retail operations, exerting a decisive influence on inventory management efficiency and profit maximization [26, 27]. As market environments become increasingly complex – particularly in the domain of perishable goods – accurately capturing demand dynamics and formulating scientifically grounded pricing strategies have emerged as focal issues for both academia and industry [2, 16, 18].

In recent years, as market conditions have continued to evolve, time series models have been widely applied to inventory management tasks. Regarding traditional time series approaches, Li *et al.* [24] employed SARIMA to forecast demand for medical devices within healthcare supply chain management; however, the study fell short in effectively handling data characterized by complex nonlinearities and long-range dependencies [24].

Keywords. KAN-LSTM, perishable vegetables, supply chain management opti-mization, demand price elasticity.

[1] School of Labor Economics, China University of Labor Relations, Beijing 100048, P.R. China.

[2] Wenlan School of Business, Zhongnan University of Economics and Law, Hubei 430073, P.R. China.

*Corresponding author: 20240001@culr.edu.cn

⋆These authors contributed equally to this work.

Mirčetić *et al.* [32] proposed and compared optimal-combination hierarchical time series forecasting methods. Through simulation and empirical analyses, they found that the optimal-combination approach delivers superior accuracy and coherence in multi-echelon supply chain networks relative to alternative methods. Nevertheless, the omission of exogenous factors limits the model's adaptability in real-world applications [32].

For deep learning-based time series methods, Gong [15] introduced an AI-driven optimization framework for cross-border e-commerce (CBEC) supply chain management that integrates neural networks, fuzzy logic, and auction theory to address the reactivity shortcomings of traditional supply chain systems; yet the system is relatively complex and demands substantial computational resources and data support [15]. Zabraoui *et al.* [40] proposed a hybrid framework combining deep learning, evolutionary search, and reinforcement learning for inventory optimization. Using the Walmart M5 dataset to benchmark multiple optimization strategies, they incorporated a genetic algorithm and a deep Q-network (GA–DQN), achieving a substantial improvement in service levels (from 61% to 94%) while reducing inventory costs. Despite its effectiveness in improving inventory accuracy, the approach's high computational complexity and stability concerns may constrain its deployment in certain real-world scenarios [40].

Long short-term memory networks (LSTM) have garnered attention for their capacity to capture, to some extent, both long-term dependencies and short-term fluctuations [17]. For example, LSTM models have achieved notable success in electric load forecasting [38], stock price prediction [14], and meteorological data analysis [11]. Nevertheless, despite strong performance across many applications, conventional LSTMs face several limitations: when dealing with time series exhibiting complex nonlinear features, they struggle to simultaneously model long-range memory and short-term volatility [22]. Their performance is particularly sensitive to input data quality and hyperparameter settings in multifactor forecasting tasks [9]. To address these issues, numerous variants have been proposed, such as BiLSTM [38], xLSTM [3], and CNN-LSTM [38], in an effort to enhance modeling of long-range dependencies and complex features. However, while these methods yield improvements in specific contexts, they do not fundamentally overcome the intrinsic limitation of traditional LSTMs in jointly capturing long-term memory and short-term fluctuations; both standard LSTMs and their variants still exhibit shortcomings in modeling intricate nonlinear dynamics.

To address these challenges, this study proposes an innovative enhanced model, KAN-LSTM, which integrates the strengths of Kolmogorov–Arnold Networks (KAN) with LSTM, thereby overcoming the limitations of conventional LSTM in processing multi-level time series data [13,19]. Leveraging KAN's nonlinear mapping capabilities, the model augments LSTM's performance in handling complex interactive features, enabling more effective capture of intricate nonlinear relationships among factors [7]. Specifically, the KAN-LSTM model not only effectively captures long-term memory but also accurately models short-term fluctuations, thereby delivering high-precision forecasts of sales and procurement prices for perishable vegetables under volatile market conditions.

Price elasticity and cross elasticity are essential theoretical tools for formulating retail pricing strategies [10]. Price elasticity reflects the sensitivity of demand to changes in a product's price, while cross elasticity measures the spillover effects of price changes among related products [8,29]. Although traditional regression models are widely used for estimating elasticity parameters, they struggle to capture complex nonlinear dynamic relationships among multiple products [5,34]. To address this, this paper employs a multivariate stepwise regression model to quantitatively analyze product price elasticity and cross elasticity, enhancing the model's interpretability and predictive power, and laying the groundwork for subsequent pricing and inventory optimization [20,37].

In terms of optimization methods, algorithms for solving nonlinear constrained problems have continued to evolve. Traditional Sequential Least Squares Programming (SLSQP) [4] performs well on constrained optimization but exhibits low computational efficiency on large-scale nonlinear problems. Global optimization algorithms such as Simulated Annealing (SA) [35], Particle Swarm Optimization (PSO) [12], Genetic Algorithms (GA) [30], and Simulated Annealing (SA) [39] demonstrate strong global search capabilities. For example, PSO excels in various engineering optimization tasks, where its particle collaboration-based global search strategy can significantly improve the efficiency of solving complex problems [12]; GA, by simulating biological evolution, is well-suited for high-dimensional nonlinear problems [28]; and SA, known for its ability to avoid local optima, has been widely applied across domains [31]. Despite their strong performance in many applications, these metaheuristic algorithms are often sensitive to initial parameters and can be inefficient when dealing with complex constraints and large-scale problems. Accordingly, this paper proposes solving with the McCormick envelope method combined with SLSQP, thereby overcoming the tendency of traditional gradient-based methods to get trapped in local optima and significantly improving solution quality and computational efficiency.

Accordingly, the main contributions of this paper can be summarized as follows:

- KAN-LSTM framework for long time series: We propose a deep learning framework based on KAN-LSTM that fuses long-term memory with short-term fluctuation information, effectively capturing complex dynamic features in time series and achieving high-accuracy forecasting for demand and procurement prices of perishable vegetable products.
- Ablation and comparative experiments validating KAN-LSTM's forecasting performance: Through systematic ablation and comparative experiments, we evaluate the forecasting performance of the KAN-LSTM model in long-horizon tasks. The results show that, for the dataset considered in this study, KAN-LSTM provides competitive performance in capturing long-range dependencies and improving prediction accuracy.
- Elasticity parameter estimation via multivariate stepwise regression: We employ multivariate stepwise regression to estimate price elasticity and cross elasticity parameters, providing theoretical support and empirical evidence for constructing pricing strategies and inventory management models.
- Construction and solution of a multi-constraint nonlinear programming model: We develop a multi-constraint nonlinear programming model for determining future order quantities and price markup rates. Under a simplified setting, the curvature properties of the objective can be analytically characterized. For the general interdependent case, we employ McCormick envelope relaxations to handle the nonlinear terms and transform the problem into a form that can be efficiently solved by SLSQP.
- Validation of the proposed optimization strategy: Through computational experiments, we verify the efficiency and stability of the proposed solution strategy in handling the multi-constraint ordering problem, demonstrating its practical effectiveness for the decision scenario considered in this study.

2. Data and methods

2.1. Data sources and description

2.1.1. Data sources

The study uses official data from real-world farmers' markets, covering six major categories of perishable vegetables: Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers. The dataset spans three years, from July 2020 to June 2023, and records transactional information. Considering that formulating ordering and pricing strategies for individual perishable vegetable items in farmers' markets is overly complex, this paper focuses on the six categories above. By analyzing key indicators such as their aggregate historical sales, wholesale prices, loss rates, and retail prices, we construct integrated forecasting models and optimization strategies to improve category-level ordering and pricing decisions.

2.1.2. Descriptive analysis

This study examines the relationship between sales and costs for perishable vegetable products, with an emphasis on sales performance and price fluctuations across different categories of perishable vegetables. By plotting line charts of sales and replenishment prices for each category, we present their temporal trends. In addition, we visualize the correlation matrix of category sales to reveal the inter-category relationships in sales.

Figure 1A illustrates the trend of replenishment prices across different categories of perishable vegetables. Overall, the replenishment prices for Aquatic Roots and Tubers are significantly higher than those of other categories and exhibit greater volatility, indicating higher uncertainty and complexity in replenishment costs. In contrast, the replenishment prices for Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, and Mushrooms are relatively stable, suggesting more consistent pricing within the supply chain for these categories.

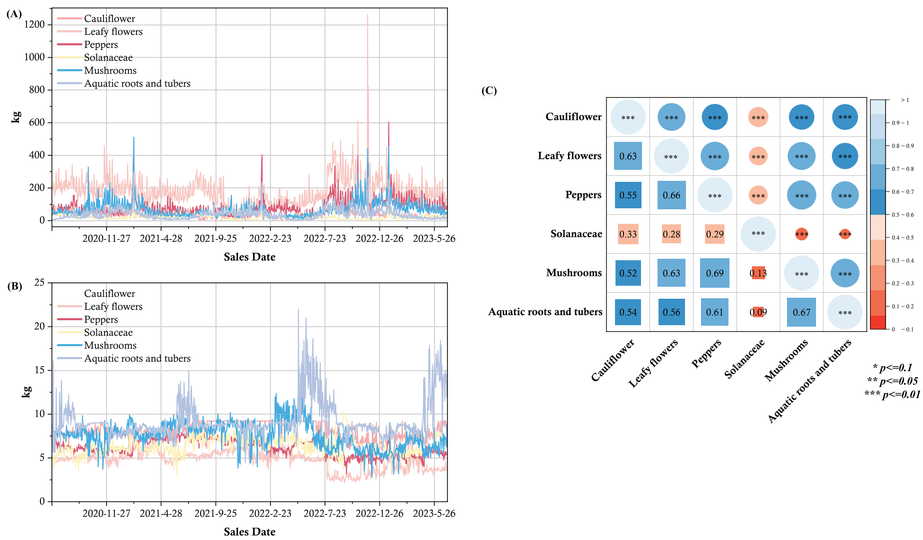


FIGURE 1. Descriptive analysis. (A) Sales trend. (B) Replenishment price trend. (C) Sales correlation coefficients.

Figure 1B shows the sales trends of each perishable vegetable category, which display pronounced cyclical fluctuations, likely related to consumer habits and supply chain cycles. Among them, Leafy Greens and Peppers have much higher sales than other categories, indicating stronger market demand, whereas Aquatic Roots and Tubers have the lowest sales, which may be associated with their higher costs.

Figure 1C presents the Pearson correlation coefficient matrix of sales across categories. The results reveal certain positive correlations among categories; for example, the correlations between Cauliflower and Leafy Greens, as well as Peppers, are relatively high ($p < 0.05$), suggesting potential linkage in consumer demand or supply chain processes. In contrast, some category pairs (such as Mushrooms and Aquatic Roots and Tubers) exhibit weaker correlations, reflecting a degree of independence in market demand.

2.2. Methods for forecasting sales and replenishment prices

2.2.1. LSTM model

Long Short-Term Memory (LSTM) networks are a special type of recurrent neural network proposed by Hochreiter and Schmidhuber in 1997, designed to address the long-term dependency problem in traditional RNNs [15].

An LSTM unit consists of several key components, including the input gate, forget gate, output gate, and cell state [15]. Through its gating mechanism, the LSTM dynamically determines, at each time step, which information to retain, which to forget, and what to output [15]. Let x_t denote the input vector at time step t, h_{t-1} the hidden state at time step $t-1$, and C_t the cell state at time step t. The updates of the LSTM components are computed as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

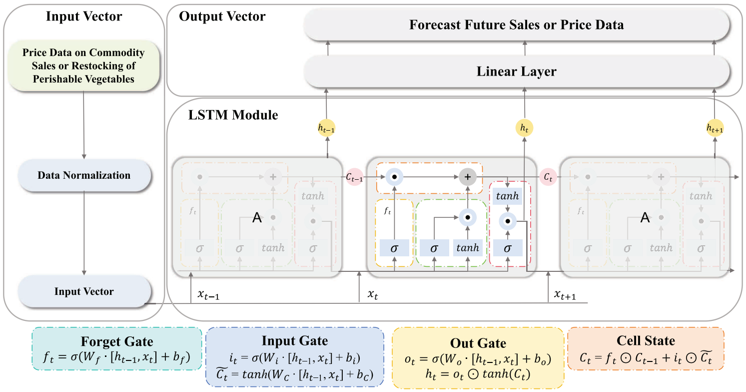


FIGURE 2. LSTM working framework.

$$h_t = o_t \odot \tanh(c_t). \tag{6}$$

Where: Equation (1) governs the forget gate, learning which historical information to retain or discard to enable effective handling of long-term memory. Equation (2) manages the input gate, determining the importance of newly added information at the current time step for the memory state. Equation (3) updates the candidate memory cell, preparing the candidate values for the main memory cell based on the current input. Equation (4) performs the actual update of the memory cell, adjusting and storing the current memory state by combining the effects of the forget and input gates. Equation (5) controls the output gate, deciding the extent of the model's response to the external environment given the memory cell. Finally, equation (6) updates the hidden state, ensuring the model can adapt to and predict future state changes when processing time series data (Fig. 2).

2.2.2. KAN-LSTM model

Kolmogorov–Arnold Networks (KAN) are a neural network architecture grounded in the Kolmogorov–Arnold representation theorem, whose core idea is to approximate any multivariate continuous function via sums and compositions of univariate functions [13, 19]. Unlike traditional neural networks that use fixed activation functions (*e.g.*, ReLU, tanh), the nonlinear functions $\phi_{q,p}(\cdot)$ in a KAN layer are learnable and can adapt during training, thereby achieving stronger nonlinear fitting capacity with a limited number of parameters. The derivation of this architecture is provided in Appendix A. To combine the nonlinear expressive power of KAN with the long- and short-term memory capability of LSTM, we propose a KAN-LSTM model. The model framework is shown in Figure 3.

(1) Input transformation via the KAN layer

The Kolmogorov–Arnold representation theorem states that any multivariate continuous function can be represented by sums and compositions of univariate functions. Even in univariate time series forecasting, the true dynamics are often highly nonlinear (*e.g.*, nonlinear demand sensitivity to price, holiday effects, and volatility clustering), which are difficult to capture with simple linear transforms or fixed activation functions. Therefore, we introduce a KAN layer at the input stage:

$$\tilde{x}_{t,q} = \sum_{p=1}^{n} \phi_{q,p}(W_{x,p} x_{t,p} + b_{x,p}), \quad x_t \in \mathbb{R}^n. \tag{7}$$

Here, x_t denotes the input feature vector at time t, representing the historical observations used for forecasting, such as sales or replenishment price data. $\phi_{q,p}(\cdot)$ denotes a learnable univariate nonlinear function based on B-splines. During training, its parameters are optimized jointly with the neural network weights $W_{x,p}$ and biases

$b_{x,p}$. Unlike fixed activation functions such as ReLU or Sigmoid, KAN adaptively learns nonlinear mappings from the data, providing greater flexibility for modeling complex temporal patterns.

In univariate time series forecasting, the input x_t is the historical observation at time t. Through a set of learnable univariate functions $\phi_{q,p}$, the KAN layer can adaptively uncover nonlinear trends, periodicities, or abrupt change patterns in the sequence, rather than being confined to the fixed activation functions used in conventional LSTM. This enables the model to exhibit greater flexibility and expressive power when modeling complex univariate time series (*e.g.*, rhythmic fluctuations in agricultural product sales).

(2) State update of RKAN (Recurrent KAN)

Extend the KAN layer to time series to form RKAN. It enhances the capture of temporal dependencies through recursive operations:

$$h_t^{\text{KAN}} = \sum_{q=1}^{2n+1} \Phi_q(\tilde{x}_{t,q}). \tag{8}$$

Here, Φ_q are learnable univariate functions. RKAN provides multiple parallel nonlinear channels to capture patterns at different time scales; for example, some channels can learn short-term daily fluctuations, while others capture long-term trends. Compared with the single tanh activation in a traditional LSTM, RKAN offers a more flexible family of functions, enabling the model to fit more complex temporal relationships with fewer parameters.

(3) Integrating LSTM's gating mechanism

To strengthen the model's ability to capture both long- and short-term dependencies, we introduce the LSTM gating mechanism to update RKAN's hidden state h_t^{KAN}. The final KAN-LSTM framework is as follows:

$$f_t = \sigma\big(W_{f,h} h_{t-1}^{\text{KAN}} + W_{f,x} x_t + b_f\big) \tag{9}$$

$$i_t = \sigma\big(W_{i,h} h_{t-1}^{\text{KAN}} + W_{i,x} x_t + b_i\big) \tag{10}$$

$$\tilde{c}_t = \tanh\big(W_{c,h} h_{t-1}^{\text{KAN}} + W_{c,x} x_t + b_c\big) \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{12}$$

$$o_t = \sigma\big(W_{o,h} h_{t-1}^{\text{KAN}} + W_{o,x} x_t + b_o\big) \tag{13}$$

$$h_t^{\text{KAN}} = o_t \odot \tanh(c_t). \tag{14}$$

Where the forget gate f_t, input gate i_t, and output gate o_t ensure that the model can selectively retain or forget information, thereby alleviating the long-term dependency problem. KAN provides richly nonlinear input features, while the LSTM gating mechanism ensures their proper propagation along the temporal dimension. The combination allows the model to express complex patterns while effectively modeling long-term dependencies.

The expressive power of a standard LSTM is constrained by its fixed activation functions (sigmoid and tanh), making it difficult to capture highly nonlinear patterns. In contrast, KAN-LSTM's learnable functions ϕ and Φ can automatically adapt to complex data structures, thereby significantly improving both fitting and generalization performance in univariate forecasting.

Although Transformers perform well in long-sequence modeling, they have high computational complexity and rely heavily on large-scale data. In the agricultural market scenario of this study, the data scale is limited and noise is substantial, making Transformers prone to overfitting. Moreover, Transformer positional encodings and attention weights are hard to map intuitively to economic meanings, whereas KAN-LSTM's functions ϕ and Φ can be interpreted as nonlinear responses of demand or price, offering greater interpretability.

2.2.3. Model evaluation metrics

To comprehensively assess the discrepancy between predictions and ground truth, this study adopts four metrics as evaluation criteria: R^2 (coefficient of determination), Mean Absolute Scaled Error (MASE), Mean

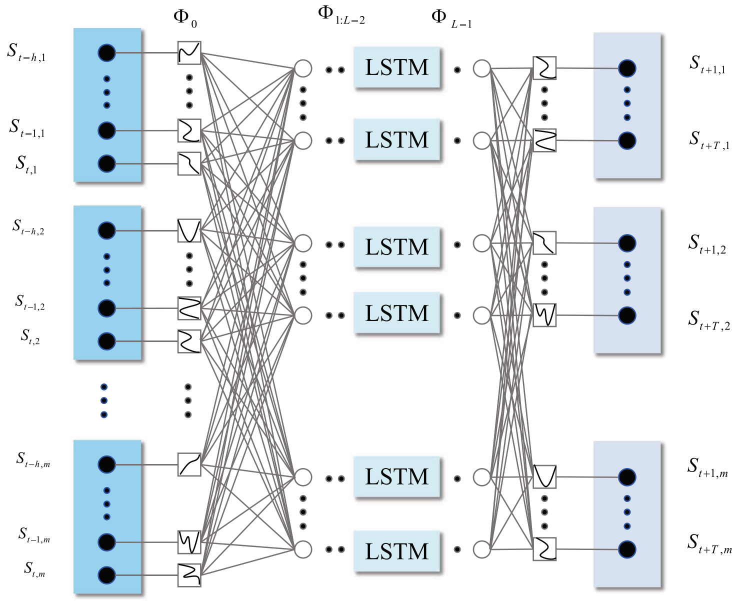


FIGURE 3. KAN-LSTM working framework.

Absolute Percentage Error (MAPE), and symmetric Mean Absolute Percentage Error (sMAPE). R^2 measures the model's ability to explain the variance in the data; values closer to 1 indicate better goodness of fit. MASE is a scale-independent metric that evaluates whether the model outperforms a "seasonal naïve forecast" by comparison; when MASE < 1, the model is superior to the benchmark. MAPE intuitively reflects the percentage of prediction error relative to the true value. Building on MAPE, sMAPE symmetrizes the denominator, effectively mitigating excessively large errors when the true value approaches zero. It provides a fairer assessment under scenarios such as low sales volume and abrupt seasonal shifts. The definitions are as follows:

$$\begin{aligned}
R^2 &= 1 - \frac{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2}, \\
\text{MASE} &= \frac{\frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t|}{\frac{1}{T-m}\sum_{t=m+1}^{T}|y_t - y_{t-m}|}, \\
\text{MAPE} &= \frac{100\%}{T}\sum_{t=1}^{T}\left|\frac{y_t - \hat{y}_t}{y_t}\right|, \\
\text{sMAPE} &= \frac{100\%}{T}\sum_{t=1}^{T}\frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2},
\end{aligned} \tag{15}$$

where y_t denotes the ground truth, $\hat{y}_t$ the prediction, $\bar{y}$ the mean of the ground truth, T the length of the time series, and m the seasonal period. In this study, we set $m = 30$ to reflect the monthly sales cycle.

2.3. Optimization and decision-making of ordering strategy

In optimizing the ordering strategy, it is necessary to comprehensively consider factors such as pricing strategy, demand forecasting, and inventory management, so as to maximize store profit and reduce operating costs. This study divides the optimization of the ordering strategy into three components: cost-plus pricing, a demand regression model, and a planning model.

2.3.1. Cost-plus pricing

Cost-plus pricing forms the basis of the ordering strategy, whose core is to determine the retail price by applying a markup to the wholesale (replenishment) cost. The pricing formula is:

$$p_i = c_i \cdot (1 + m_i). \tag{16}$$

Here, p_i denotes the selling price of product i, c_i the replenishment cost of product i, and m_i the markup rate of product i.

A reasonable markup rate m_i is a key decision variable for the subsequent planning model. To determine the optimal markup, it must be jointly optimized in conjunction with the demand regression model and planning objectives.

2.3.2. Demand regression model

The demand D_i for a product is influenced not only by its own selling price p_i but also by the prices p_j and sales D_j of other related products, typically exhibiting nonlinear relationships. To comprehensively analyze price elasticity of demand and cross-sales elasticity, this study constructs a multivariate demand regression model based on demand elasticity theory. By fitting historical sales data with multiple product prices and sales, we obtain the following log-demand regression:

$$\ln(D_i) = \alpha - \beta_i \ln(p_i) + \sum_j \gamma_j \ln(p_j) + \sum_j \delta_j \ln(D_j). \tag{17}$$

Where D_i is the demand for the target product, p_i is the selling price of the target product, and p_j, D_j, γ_j, and δ_j respectively denote the selling prices and sales of other related products, the cross-price elasticity coefficients, and the cross-sales elasticity coefficients; α is the scale parameter, and β_i is the own-price elasticity coefficient. Combining the cost-plus pricing formula $p_i = c_i \cdot (1 + m_i)$, demand can be further expressed as:

$$D_i = e^{\alpha} \cdot p_i^{-\beta_i} \cdot \prod_j p_j^{\gamma_j} \cdot \prod_j D_j^{\delta_j}. \tag{18}$$

By estimating the parameters α, β, γ_j, and δ_j, we can characterize in detail how the demand for the target product responds to changes in its own price, the prices of related products, and their sales, thereby providing a scientific basis for optimizing pricing and ordering strategies.

2.3.3. Construction of a nonlinear programming model

Building on cost-plus pricing and the demand regression model, we construct a planning model that takes the order quantity Q_i and the markup rate m_i as decision variables, aiming to maximize store profit while controlling inventory costs. The objective function and constraints are as follows:

(1) Objective function

The store's total profit consists of sales profit and inventory costs, expressed as:

$$\max_{Q,m} \text{Profit} = \sum_i [(p_i - c_i) \cdot D_i - h_i \cdot (Q_i - D_i)]. \tag{19}$$

Here, $(p_i - c_i) \cdot D_i$ is the sales profit of product i; $h_i \cdot (Q_i - D_i)$ is the inventory holding cost, where h_i is the spoilage rate of vegetable product i, and Q_i is the order quantity of product i.

Table 1. Hyperparameter settings.

Parameter	Value	Parameter	Value
Number of hidden layers	2	Optimizer	Adam
Hidden units per layer	6	Learning rate	0.001–0.005
Fully connected layer neurons	7	Loss function	Mean Squared Error
KAN nonlinear basis	B-splines	Training epochs	400

(2) Constraints

Constraint 1: Demand-satisfaction constraint. The order quantity must at least meet demand, *i.e.*, inventory is nonnegative.

$$Q_i \geq e^{\alpha_i} p_i^{-\beta_i} \prod_j p_j^{\gamma_j} \prod_j D_j^{\delta_j}, \qquad \forall i. \tag{20}$$

Constraint 2: Markup bounds. The markup rate must lie within a reasonable range.

$$m_{\min} \leq m_i \leq m_{\max}, \qquad \forall i. \tag{21}$$

(3) Mathematical form of the optimization model

Combining the objective function and the constraints above, the planning model is obtained as:

$$\begin{aligned}
&\max_{Q,m} \text{Profit} = \sum_i [(p_i - c_i) \cdot D_i - h_i \cdot (Q_i - D_i)],\\
&\text{subject to:}\\
&Q_i \geq e^{\alpha_i} p_i^{-\beta_i} \prod_j p_j^{\gamma_j} \prod_j D_j^{\delta_j}, \quad \forall i,\\
&m_{\min} \leq m_i \leq m_{\max},\\
&D = e^{\alpha} \cdot [c \cdot (1+m)]^{-\beta_i} \cdot \prod_j p_j^{\gamma_j} \cdot \prod_j D_j^{\delta_j},\\
&p_i = c_i \cdot (1 + m_i).
\end{aligned} \tag{22}$$

3. Computational results

3.1. KAN-LSTM forecasting results

3.1.1. Parameter settings

This study constructs a KAN-LSTM model using historical sales data for six categories of perishable vegetables, including cauliflower and others. During model training, we first trained according to the hyperparameter configuration shown in Table 1. To enhance the robustness and generalization of model evaluation, five-fold temporal cross-validation was employed to assess performance. Each model was run 10 times, and the mean and 95% confidence interval of the performance metrics were computed. After cross-validation, the model was retrained on the full dataset using the same optimal hyperparameters to fully leverage all historical information and to support subsequent sales forecasting.

3.1.2. Iterative curves

Subsequently, based on the specified hyperparameters, five-fold temporal cross-validation training was conducted separately for each dataset, and the evolution of training and validation loss values across epochs was

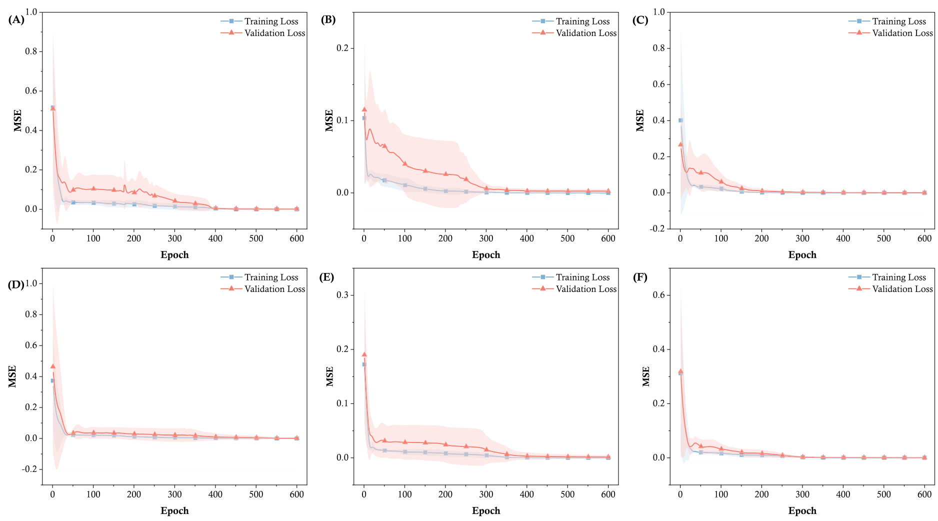


FIGURE 4. Five-fold cross-validation loss curves of the KAN-LSTM model on six vegetable sales datasets.

recorded. To intuitively illustrate the convergence characteristics, loss curves with mean values and one standard-deviation shaded bands were plotted (see Figs. 4 and 5), reflecting the model's stability and consistency across folds.

Figures 4A–4F correspond to predictions for Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers, respectively. The shaded bands indicate the range of one standard deviation.

As shown in the Figure 4, the training and validation loss curves under five-fold cross-validation for the KAN-LSTM model on each vegetable replenishment price dataset exhibit sustained decreases and gradual stabilization, indicating that the model effectively learns temporal dependencies in the price sequences. The rates of decline and the amplitudes of fluctuations vary across categories, reflecting differences in price-volatility characteristics among datasets. Overall, the convergence trends of the training and validation curves are consistent, and the one-standard-deviation shaded bands are relatively narrow, suggesting high stability across folds, good generalization capability, satisfactory convergence, and no evident overfitting.

From (A) to (F), the Figure 5 correspond to Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers, respectively. The shaded bands indicate one standard deviation.

From the figure, the training and validation loss curves under five-fold cross-validation for each dataset generally exhibit a gradual decline with increasing epochs, and the model enters a stabilized phase after relatively few epochs. This indicates that the KAN-LSTM model quickly captures the key features of the time series and achieves effective fitting. The magnitudes of loss variation differ across vegetable categories, which is related to the volatility and temporal dependence characteristics of each dataset. Overall, the gaps between the training and validation curves under five-fold cross-validation are small, and the shaded bands converge well, suggesting high stability across folds and no evident overfitting.

3.1.3. Model evaluation and forecasting results

After completing model training and convergence verification, we systematically evaluated the predictive performance of the KAN-LSTM model on different datasets by conducting five-fold temporal cross-validation on the sales and replenishment price data for the six vegetable categories. Each dataset experiment was run independently 10 times to mitigate the impact of random initialization, and the mean and 95% confidence

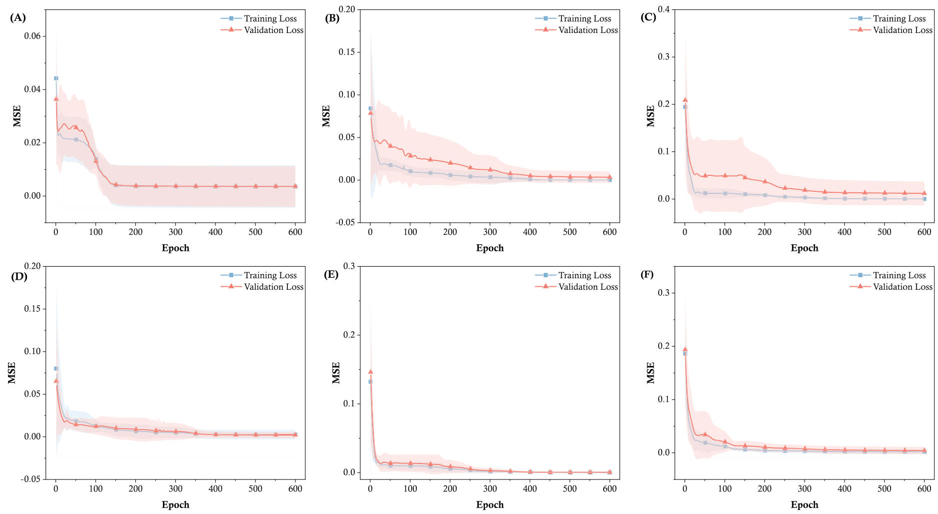


FIGURE 5. Five-fold cross-validation loss curves of the KAN-LSTM model on six vegetable replenishment price datasets.

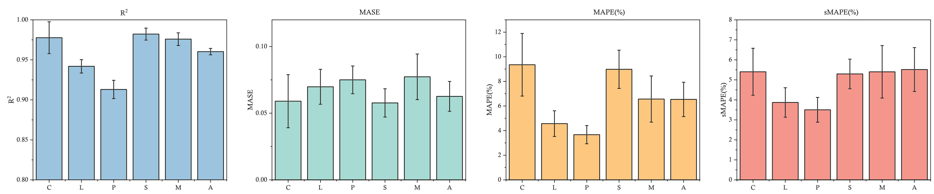


FIGURE 6. Five-fold cross-validation evaluation of the KAN-LSTM model on the replenishment price datasets. C, L, P, S, M, and A denote Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers, respectively.

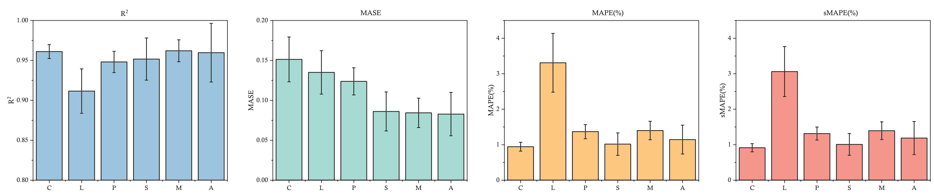


FIGURE 7. Five-fold cross-validation evaluation of the KAN-LSTM model on the sales datasets. C, L, P, S, M, and A denote Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers, respectively.

interval were computed across runs to enhance robustness and statistical reliability. Figure 6 presents the cross-validation results on the sales datasets, while Figure 7 shows the corresponding performance on the replenishment price datasets.

As shown in Figures 6 and 7, the KAN-LSTM model demonstrates high fitting accuracy and good stability across both the sales and replenishment price datasets for all six vegetable categories. For the sales forecasting task, R^2 remains between 0.93 and 0.98, with Cauliflower at 0.9777 ($\pm$0.0199) and Mushrooms at 0.9759

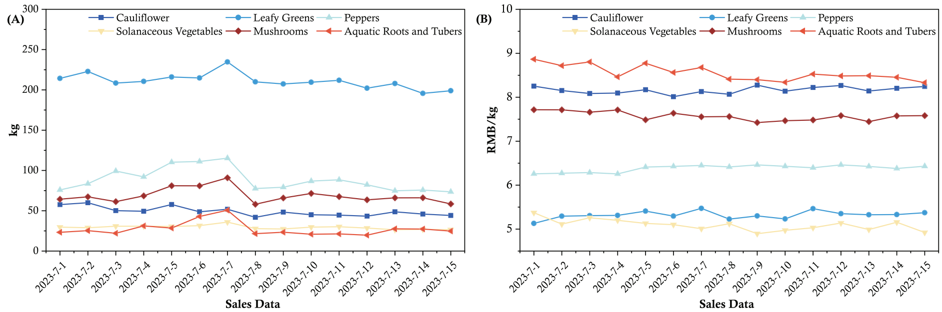


FIGURE 8. Forecasting results. (A) Sales forecasts; (B) Replenishment price forecasts.

(±0.0079), indicating strong capability in capturing price dynamics. All sMAPE and MAPE values are within 8%, suggesting similarly high accuracy in price prediction. For replenishment price forecasting, R^2 exceeds 0.90 for all datasets, with Cauliflower and Leafy Greens reaching 0.9611 (±0.0087) and 0.9116 (±0.0278), respectively, indicating that the model effectively captures the temporal characteristics of sales variations. All MASE values are below 1.0, and sMAPE and MAPE are around 3%, reflecting small prediction errors and good model fit. Overall, the metrics exhibit narrow variability and short 95% confidence-interval error bars in both tasks, indicating high consistency across 10 independent runs and five-fold cross-validation, with strong generalization and stable time-series forecasting capability. The model's performance in forecasting future sales and price trends is shown in Figure 8.

3.2. Ordering strategy

3.2.1. Results of the demand regression model

For the demand regression model, this study employs a stepwise regression approach to solve the nonlinear specification. By estimating the regression coefficients of each variable and their significance levels (P-values), we screen explanatory variables that have a significant impact on demand fluctuations [25]. The parameter estimates and significance results of the final regression models are presented in Table 2.

From Table 2, all models exhibit good overall fit (R^2 between 0.567 and 0.853, with relatively high F-statistics), indicating that the selected explanatory variables jointly have strong explanatory power for demand volatility. The own-price terms for each category (*e.g.*, Model I $p_C = -0.986^{***}$, Model II $p_L = -0.657^{***}$, Model III $p_P = -0.599^{***}$, $p_M = -0.346^{***}$) are significantly negative, showing typical price inelasticity characteristics. Meanwhile, several cross-price terms are positive (*e.g.*, Model IV $p_C = 0.573^{***}$, Model V $p_P = 0.501^{***}$), reflecting co-movement due to substitution preferences among different categories. Most cross-sales terms are significantly positive (*e.g.*, Model I $D_L = 0.491^{***}$, $D_M = 0.420^{***}$), indicating mutually reinforcing effects across multi-category demand. In summary, the significant negative effect of own price, together with the mutually reinforcing cross-price and cross-sales effects, jointly explains the demand structure of each category. Based on the integrated forecasting results, the nonlinear programming model in this paper is formulated as:

$$\max_{Q,m} \text{Profit} = \sum_i [(p_i - c_i) \cdot D_i - h_i \cdot (Q_i - D_i)], \tag{23}$$

TABLE 2. Parameter estimates and significance results.

	Model I	Model II	Model III	Model IV	Model V	Model VI
Intercept term	*const* (2.189**)	*const* (3.120***)	*const* (3.278***)	*const* (-2.639***)	*const* (-0.413)	*const* (-0.943)
Variable 1	p_C (-0.986***)	p_L (-0.657***)	p_P (-0.599***)	p_C (0.573***)	p_M (-0.346***)	p_L (-1.581***)
Variable 2	p_S (0.591**)	p_C (0.306**)	p_M (0.270**)	D_P (0.675***)	p_P (0.501***)	p_S (0.797**)
Variable 3	D_M (0.420***)	D_P (0.272**)	D_M (0.510***)	D_M (0.411**)	D_P (0.716***)	D_P (1.046***)
Variable 4	D_L (0.491***)	D_M (0.348***)	D_S (0.159***)		D_L (0.345***)	
Variable 5		D_C (0.207***)	D_C (0.107**)			
R^2	0.771	0.787	0.853	0.638	0.848	0.567
F-statistics	47.246***	40.530	63.643	33.514	78.172	24.912

Notes: Model I to VI denote the six specifications. Values in parentheses indicate variable coefficients; significance levels ***, **, and * denote significance at the 1%, 5%, and 10% levels, respectively.

$$
\begin{aligned}
&\text{subject to:}\\
&Q_i \geq e^{\alpha_i} p_i^{-\beta_i} \prod_j p_j^{\gamma_j} \prod_j D_j^{\delta_j}, \qquad \forall i,\\
&m_{\min} \leq m_i \leq m_{\max},\\
&D_C = e^{2.189} \cdot p_C^{-0.986} \cdot p_S^{0.591} \cdot D_M^{0.420} \cdot D_L^{0.491},\\
&D_L = e^{3.120} \cdot p_L^{-0.657} \cdot p_C^{0.306} \cdot D_P^{0.272} \cdot D_M^{0.348} \cdot D_C^{0.207},\\
&D_P = e^{3.278} \cdot p_P^{-0.599} \cdot p_M^{0.270} \cdot D_M^{0.510} \cdot D_S^{0.159} \cdot D_C^{0.107},\\
&D_S = e^{-2.969} \cdot p_C^{0.573} \cdot D_P^{0.675} \cdot D_M^{0.411},\\
&D_M = e^{-0.413} \cdot p_M^{-0.346} \cdot p_P^{0.501} \cdot D_P^{0.716} \cdot D_L^{0.345},\\
&D_A = e^{-0.943} \cdot p_L^{-1.581} \cdot p_S^{0.797} \cdot D_P^{1.046},\\
&p_i = c_i \cdot (1 + m_i).
\end{aligned}
\tag{24}
$$

Here, D_C, D_L, D_P, D_S, D_M and D_A denote the sales volumes of Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers, respectively. P_C, P_L, P_P, P_S, P_M and P_A denote the replenishment prices (including markup rate) for Cauliflower, Leafy Greens, Peppers, Solanaceous Vegetables, Mushrooms, and Aquatic Roots and Tubers, respectively.

3.2.2. Solving the ordering decisions for the next 15 days in the farmers' market

To better understand the structure of the optimization problem, we distinguish between a simplified case and the general interdependent case. In the simplified setting, analytical curvature conditions can be derived for the one-dimensional markup term. In the general case, however, price and demand interdependencies introduce coupled nonlinear terms, making a direct proof of global convexity challenging. Therefore, rather than claiming global convexity for the full model, we employ McCormick envelopes to construct convex relaxations of the nonlinear terms and solve the resulting problem with the SLSQP algorithm.

The Case 1: Simplified scenario. In the simplified scenario, let $\gamma_j = \delta_j = 0$, *i.e.*, the recursive terms $p_j^{\gamma_j}$ and $D_j^{\delta_j}$ are omitted. The objective term associated with the markup decision can then be written as:

$$f_i(m_i) = (c_i m_i + h_i) e^{\alpha} p_i^{-\beta_i}. \tag{25}$$

For this simplified objective, differentiating with respect to m_i yields a second-derivative condition characterized by the following expression:

$$S(m_i) = (1 - \beta_i) m_i - (1 + \beta_i) \frac{h_i}{c_i} + 2. \tag{26}$$

The sign pattern of $S(m_i)$ characterizes the local curvature of the simplified one-dimensional objective term and shows how the markup decision interacts with the elasticity parameter β_i and the spoilage-cost ratio h_i/c_i. This analytical result provides theoretical insight into the structure of the simplified problem, but it is not used to claim global convexity for the full interdependent optimization model.

The Case 2: General interdependent scenario. In the general scenario, the recursive terms $p_j^{\gamma_j}$ and $D_j^{\delta_j}$ are retained, so the demand function contains coupled nonlinear interactions among prices and demands:

$$D_i = e^{\alpha} p_i^{-\beta_i} \prod_j p_j^{\gamma_j} \prod_j D_j^{\delta_j}. \tag{27}$$

Due to the potential non-convexity introduced by price and demand interdependencies, a direct proof of global convexity is challenging in this general case. To illustrate the complexity of the objective landscape, we perform a numerical Hessian eigenvalue analysis. The Hessian results are used only as a local curvature diagnostic: they indicate whether sampled regions are locally convex, locally concave, or indefinite, but they do not establish global convexity of the original optimization problem.

Given this potential non-convexity, we adopt the McCormick envelope method to handle the nonlinear interaction terms. Specifically, McCormick envelopes introduce auxiliary variables and bounding constraints to construct convex relaxations of the nonlinear products, thereby transforming the problem into a form amenable to gradient-based solvers such as SLSQP.

Consider common product terms in the objective and constraints, such as $p_j^{\gamma_j} \times D_j^{\delta_j}$. Under the McCormick envelope framework, each such product is replaced by an auxiliary variable together with corresponding envelope constraints. To apply this method, bounds on p_j and D_j are required. Suppose their bounds are:

$$p_j^L < p_j < p_j^U, \quad D_j^L < D_j < D_j^U. \tag{28}$$

These bounds can be obtained from market data or expert estimates. For each product term $z_j = p_j^{\gamma_j} \times D_j^{\delta_j}$, the McCormick envelope provides the following bounding constraints to ensure that the auxiliary variable z_j remains within the admissible range:

$$z_j \geq \min\left(p_j^L \times D_j^L, p_j^L \times D_j^U, p_j^U \times D_j^L, p_j^U \times D_j^U\right) \tag{29}$$

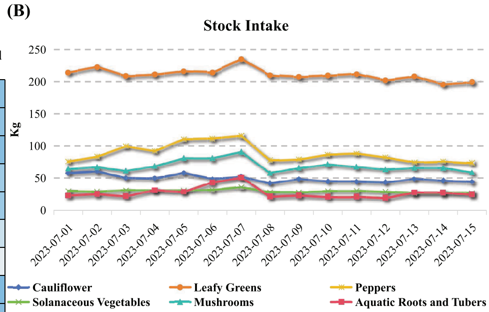
$$z_j \leq \max\left(p_j^L \times D_j^L, p_j^L \times D_j^U, p_j^U \times D_j^L, p_j^U \times D_j^U\right). \tag{30}$$

These constraints restrict z_j within the upper and lower bounds implied by $p_j^{\gamma_j} \times D_j^{\delta_j}$, thereby improving the tractability of the optimization problem in the general interdependent setting.

(A)

Date	Cauliflower	Leafy flowers	Peppers	Solanaceae	Mushrooms	Aquatic roots and tubers
2023-7-1	0.4690	0.5000	0.4688	0.3899	0.4741	0.4015
2023-7-2	0.4723	0.5000	0.4810	0.3870	0.4796	0.4052
2023-7-3	0.4514	0.5000	0.5000	0.3906	0.4674	0.3986
2023-7-4	0.4499	0.5000	0.4941	0.3900	0.4820	0.4160
2023-7-5	0.4680	0.5000	0.5000	0.3891	0.5000	0.4125
2023-7-6	0.4475	0.5000	0.5000	0.3902	0.5000	0.4419
2023-7-7	0.4554	0.5000	0.5000	0.3950	0.5000	0.4600
2023-7-8	0.4345	0.5000	0.4744	0.3856	0.4598	0.3954
2023-7-9	0.4501	0.5000	0.4779	0.3835	0.4722	0.3990
2023-7-10	0.4416	0.5000	0.4896	0.3869	0.4836	0.3936
2023-7-11	0.4416	0.5000	0.4913	0.3880	0.4762	0.3954
2023-7-12	0.4395	0.5000	0.4828	0.3867	0.4705	0.3918
2023-7-13	0.4490	0.5000	0.4702	0.3830	0.4729	0.4085
2023-7-14	0.4441	0.5000	0.4708	0.3846	0.4753	0.4077
2023-7-15	0.4410	0.5000	0.4682	0.3827	0.4608	0.4016

(B)



(C)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5
Profit	1235.06	1316.16	1263.17	1297.18	1452.32
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10
Profit	1461.37	1637.25	1139.67	1200.16	1224.24
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15
Profit	1248.42	1171.87	1203.92	1169.33	1131.60

FIGURE 9. Decision results for the next 15 days in the farmers' market. (A) Pricing markup decisions. (B) Purchasing quantity decisions. (C) Daily profit of the farmers' market.

3.3. Solution results

To determine the purchase quantities and markups for the next 15 days, this study leverages the demand forecasting model and objective function, and performs optimization using the McCormick envelope method combined with the SLSQP algorithm [4]. Constraints are imposed to ensure solution feasibility, with the goal of maximizing total profit. The results are shown in Figure 9.

The optimization results indicate that the markup rates range from 0.38 to 0.50. Among them, Solanaceous vegetables and aquatic roots and tubers have relatively lower markups, while leafy greens are higher. The daily purchase quantity decisions fluctuate, with leafy greens having the highest purchase volume and aquatic roots and tubers the lowest. The daily profit ranges from 1169.33 to 1461.37 yuan, with a total profit of 19 151.72 yuan. Overall, profits show a fluctuating upward trend, validating the effectiveness of the optimization strategy.

4. DISCUSSION AND COMPARISON

4.1. Ablation study on long-term memory in time series

To evaluate the long-term memory capability of the KAN-LSTM model, we tested different data window lengths, including 1-year, 2-year, and 3-year windows, and conducted comparative experiments with multiple models such as LSTM, xLSTM, TCN, Transformer, and CNN-Transformer. To ensure accurate assessment and model generalization, each experiment was run 10 times with five-fold cross-validation. Moreover, since the dataset contains six vegetable categories, for each time window we aggregated the prediction results across the six categories, computing the mean and standard deviation over all runs for each model under that window. This yields more comprehensive and stable evaluation metrics.

In sales forecasting, Table 3 presents performance comparisons between the KAN-LSTM model and other models across different time windows (365 days, 730 days, and 1095 days). The results show that KAN-LSTM performs strongly across all evaluation metrics, particularly in R^2, MASE, MAPE, and sMAPE. Specifically, under the 365-day window, KAN-LSTM achieves an R^2 of 0.8858, MASE of 0.0903, MAPE of 8.6640, and sMAPE of 5.6219. Under the 730-day and 1095-day windows, KAN-LSTM's performance further improves, reaching R^2 values of 0.9447 and 0.9585, MASE of 0.0727 and 0.0669, MAPE of 6.2943 and 6.6128, and sMAPE

Table 3. Ablation comparison of KAN-LSTM across different time windows for sales forecasting.

Models	Window	R^2		MASE		MAPE		sMAPE	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
KAN-LSTM [13]	365	**0.8858**	0.0492	**0.0903**	0.0456	**8.6640**	4.8044	**5.6219**	3.1203
	730	**0.9447**	0.0316	**0.0727**	0.0256	**6.2943**	5.4679	**4.7846**	3.0393
	1095	**0.9585**	0.0289	**0.0669**	0.0212	**6.6128**	3.0884	**4.8317**	1.5986
LSTM [17]	365	0.6221	0.0525	0.3514	0.0378	29.2831	9.2310	20.2814	3.1241
	730	0.6943	0.0527	0.3015	0.0365	25.4332	8.4562	18.6350	3.3978
	1095	0.7362	0.0613	0.2681	0.0509	20.8095	10.8396	13.6622	3.6938
xLSTM [3]	365	0.8090	0.0655	0.1649	0.0490	13.5229	8.4598	8.9272	3.6890
	730	0.7356	0.0638	0.2163	0.0559	17.1387	10.7751	12.7759	4.2535
	1095	0.7630	0.0705	0.2075	0.0562	18.8210	7.0131	13.1914	4.3893
TCN [23]	365	0.7856	0.0642	0.1957	0.0337	20.8326	8.3669	14.8660	3.3945
	730	0.8114	0.0574	0.1799	0.0433	13.8645	9.7097	8.2518	4.9006
	1095	0.8207	0.0652	0.1366	0.0531	12.3527	10.1445	8.3466	4.4981
Transformer [36]	365	0.7369	0.0693	0.2358	0.0426	20.7238	6.8548	16.2435	3.8009
	730	0.7669	0.0592	0.2009	0.0440	18.0082	11.1339	14.4270	5.6564
	1095	0.8244	0.0697	0.1352	0.0650	16.5298	12.5055	13.0864	7.7784
CNN-Transformer [41]	365	0.7591	0.0614	0.2736	0.0527	19.0598	12.1490	14.1433	4.4618
	730	0.8237	0.0681	0.1403	0.0494	11.3362	8.4267	11.6209	5.9797
	1095	0.8535	0.0669	0.1019	0.0571	10.4202	6.8513	8.0388	4.1411

Table 4. Ablation comparison of KAN-LSTM across different time windows for replenishment price forecasting.

Models	Window	R^2		MASE		MAPE		sMAPE	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
KAN-LSTM [13]	365	**0.8935**	0.0472	**0.1501**	0.0403	**2.0043**	0.5988	**2.0865**	0.6528
	730	**0.9069**	0.0360	**0.1227**	0.0403	**1.9374**	0.9910	**1.8667**	0.8994
	1095	**0.9191**	0.0370	**0.1106**	0.0434	**1.5305**	1.0051	**1.4805**	0.9084
LSTM [17]	365	0.6937	0.0684	0.4322	0.0499	6.3196	0.9702	6.2855	1.0508
	730	0.7416	0.0637	0.2906	0.0610	4.0453	1.7162	3.7819	1.4487
	1095	0.7098	0.0767	0.3284	0.0806	5.2385	1.4454	5.5066	1.2413
xLSTM [3]	365	0.7820	0.0790	0.2164	0.0502	2.8076	1.0799	2.8026	1.0670
	730	0.7273	0.0699	0.2685	0.0742	3.9735	1.7996	3.7983	1.5892
	1095	0.7094	0.0699	0.3009	0.0698	4.0865	1.4949	4.0746	1.3199
TCN [23]	365	0.8686	0.0643	0.1630	0.0432	2.3110	0.7698	2.2242	0.7301
	730	0.8732	0.0751	0.1934	0.0840	2.0331	1.7211	2.0623	1.5481
	1095	0.8540	0.0696	0.2014	0.0603	2.6222	1.8941	2.3729	1.5736
Transformer [36]	365	0.7252	0.0714	0.3246	0.0570	3.9490	0.8832	3.9164	0.8990
	730	0.7333	0.0710	0.3145	0.0595	3.7925	2.3715	3.9937	2.1184
	1095	0.7501	0.0634	0.2939	0.0893	3.2087	1.7820	3.1001	1.6195
CNN-Transformer [41]	365	0.7320	0.0798	0.3223	0.0686	3.9547	0.8298	3.9491	0.8731
	730	0.7508	0.0702	0.2971	0.0640	3.2898	2.2872	3.1934	2.0604
	1095	0.7949	0.0774	0.2597	0.0887	3.0886	1.9112	3.0964	1.7030

TABLE 5. Hypothesis testing results for sales and replenishment price forecasting performance.

Model	Time window	Sales dataset		Restocking price dataset	
		t-statistic	p-value	t-statistic	p-value
	365	28.3891	0.0000	18.6228	0.0000
LSTM [17]	730	31.5648	0.0000	17.4993	0.0000
	1095	25.4081	0.0000	19.0379	0.0000
	365	7.2618	0.0000	9.3851	0.0000
xLSTM [3]	730	22.7493	0.0000	17.6936	0.0000
	1095	19.8749	0.0000	20.5381	0.0000
	365	9.5957	0.0000	2.4181	0.0187
TCN [23]	730	15.7583	0.0000	3.1344	0.0027
	1095	14.9667	0.0000	6.3974	0.0000
	365	13.5709	0.0000	15.2311	0.0000
Transformer [36]	730	20.5233	0.0000	16.8921	0.0000
	1095	13.7665	0.0000	17.8331	0.0000
	365	12.4734	0.0000	13.4928	0.0000
CNN-Transformer [41]	730	12.4844	0.0000	15.3265	0.0000
	1095	11.1605	0.0000	11.2141	0.0000

of 4.7846 and 4.8317, respectively. In contrast, other models such as LSTM, xLSTM, TCN, Transformer, and CNN-Transformer generally show weaker performance than KAN-LSTM across the evaluated windows. For example, the LSTM model under the 365-day window yields an R^2 of 0.6221 and MAPE of 29.2831, noticeably lower than that of KAN-LSTM under the same window.

For replenishment price forecasting in Table 4, the KAN-LSTM model also significantly outperforms other models, especially on R^2 and MAPE. Under the 365-day window, KAN-LSTM achieves an R^2 of 0.8935 and MAPE of 2.0043; under the 730-day and 1095-day windows, R^2 further increases to 0.9069 and 0.9191, while MAPE decreases to 1.9374 and 1.5305, respectively. By comparison, the LSTM model under the 365-day window attains only an R^2 of 0.6937 and MAPE of 6.3196, clearly lower than KAN-LSTM's performance.

While newer architectures such as efficient Transformers have shown strong performance in large-scale sequence modeling tasks, the KAN-LSTM adopted in this study provides a favorable balance between nonlinear representation capacity and manageable model complexity for our midsized agricultural dataset, making it an effective and well-matched choice for this specific forecasting problem. Its learnable nonlinear components may also offer a useful path toward interpreting market response patterns in future work. By computing the mean and standard deviation for each model under different windows, we further conduct hypothesis testing to examine whether KAN-LSTM yields consistently better sMAPE results than the compared models, as shown in Table 5.

Table 5 presents the hypothesis testing results comparing KAN-LSTM with other models on the sales forecasting and replenishment price forecasting tasks. The results indicate that KAN-LSTM significantly outperforms the other models across all time windows. In the sales forecasting task, KAN-LSTM's t-values are far higher than those of the other models, with p-values all equal to 0.0000, highlighting its advantage in long-term memory capability. The replenishment price forecasting results are similarly significant: KAN-LSTM's p-values are 0.0000 across all time windows, indicating more stable performance on this task within the present experimental setting. Compared with other models, while LSTM and xLSTM perform well in some time windows, their t-values and p-values are notably inferior to those of KAN-LSTM, especially under longer time windows. Overall, KAN-LSTM exhibits stronger stability and predictive accuracy when handling forecasting tasks across different time windows.

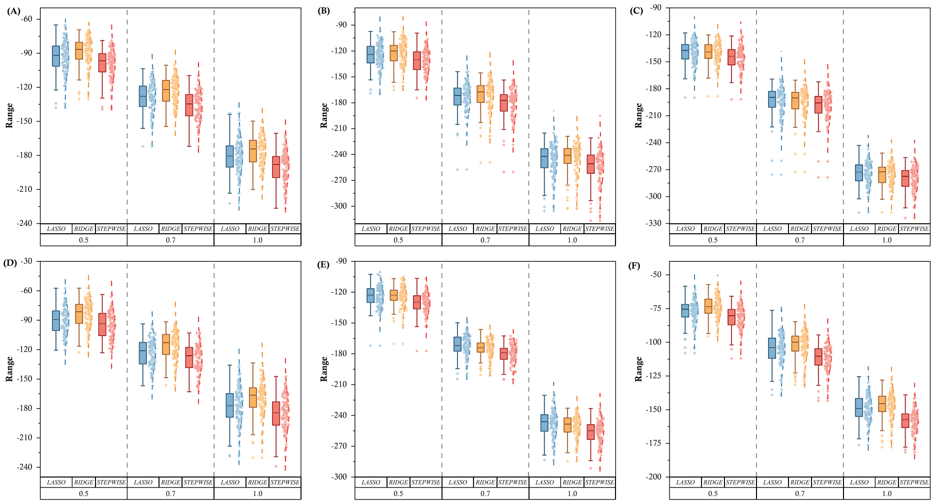


FIGURE 10. Comparison results of demand regression models.

4.2. Comparison of demand regression models

In the comparison of demand regression models, we adopted three methods: stepwise regression, LASSO regression, and ridge regression, and performed multiple resampling of the time series data using the Moving Block Bootstrap (MBB). The MBB method generates samples of varying lengths by randomly drawing multiple consecutive data blocks from the time series, thereby simulating different data scenarios. For each resample, we use the Bayesian Information Criterion (BIC) to evaluate model performance. The BIC is given by:

$$\mathrm{BIC} = n \log\left(\frac{\mathrm{RSS}}{n}\right) + k \log(n) \tag{31}$$

where n is the sample size, RSS is the residual sum of squares, and k is the number of model parameters. BIC balances model goodness-of-fit and complexity, with smaller BIC values indicating a better model. Using this approach, we can comprehensively compare the performance of different models under different data sampling ratios, with results shown in Figure 10.

Figure 10 presents the comparative results of the demand regression models under different methods. The stepwise regression method can effectively select variables that are significantly associated with demand fluctuations and eliminate redundant features, preventing model overfitting. In contrast, although LASSO and ridge regression provide regularization, they still have limitations in feature selection and complexity control, especially when inter-variable effects are relatively complex. Therefore, in this study, stepwise regression is the more suitable choice. By applying the BIC criterion to screen for the optimal model, it provides more accurate and stable forecasting results.

4.3. Discussion of ordering strategies

To further analyze the advantages of solving this paper's non-convex optimization problem using the McCormick envelope method combined with SLSQP, we selected several global optimization methods – Simulated Annealing (SA) [35], Particle Swarm Optimization (PSO) [21], Genetic Algorithm (GA) [1], and Tabu Search (TS) [33] – and conducted comparative analyses against the McCormick envelope + SLSQP approach. Algorithm parameters are provided in Appendix A. By optimizing the inbound quantities and markups for the

(A)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5	**Solution Time**
Profit	1235.06	1316.16	1263.17	1297.18	1452.32	0.20 Seconds
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10	
Profit	1461.37	1637.25	1139.67	1200.16	1224.24	**Total Profit**
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15	19151.72 yuan
Profit	1248.42	1171.87	1203.92	1169.33	1131.60	

(B)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5	**Solution Time**
Profit	1235.06	1316.16	1263.17	1297.18	1452.32	4.07 Seconds
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10	
Profit	1461.37	1637.25	1139.67	1200.16	1224.24	**Total Proft**
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15	19151.72 yuan
Profit	1248.42	1171.87	1203.92	1169.33	1131.60	

(C)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5	**Solution Time**
Profit	1235.06	1316.16	1263.17	1297.18	1452.32	12.52 Seconds
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10	
Profit	1461.37	1637.25	1139.67	1200.16	1224.24	**Total Profit**
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15	19151.72 yuan
Profit	1248.42	1171.87	1203.92	1169.33	1131.60	

(D)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5	**Solution Time**
Profit	1226.79	1311.80	1261.35	1294.62	1446.59	7.74 Seconds
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10	
Profit	1451.09	1632.53	1136.21	1192.29	1218.25	**Total Proft**
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15	19066.73 yuan
Profit	1245.44	1163.99	1195.70	1160.59	1129.46	

(E)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5	**Solution Time**
Profit	1234.87	1315.65	1263.01	1296.94	1452.13	1644.09 Seconds
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10	
Profit	1461.25	1637.05	1139.58	1200.03	1224.14	**Total Proft**
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15	19148.56 yuan
Profit	1248.09	1171.48	1203.56	1169.25	1131.53	

(F)

Date	2023-7-1	2023-7-2	2023-7-3	2023-7-4	2023-7-5	**Solution Time**
Profit	1220.58	1297.24	1253.31	1282.13	1435.84	0.16 Seconds
Date	2023-7-6	2023-7-7	2023-7-8	2023-7-9	2023-7-10	
Profit	1451.03	1611.45	1131.74	1196.28	1210.65	**Total Proft**
Date	2023-7-11	2023-7-12	2023-7-13	2023-7-14	2023-7-15	18967.33 yuan
Profit	1236.99	1164.20	1196.93	1163.93	1115.02	

FIGURE 11. Optimization results. Panels (A)–(F) correspond to the results of the McCormick envelope + SLSQP method, the McCormick envelope + spatial branch-and-bound method, SA, PSO, GA, and TS, respectively.

next 15 days, we evaluate differences in optimization outcomes among algorithms when tackling multi-constraint nonlinear programming problems. The results are shown in Figure 11.

According to Figure 11, the McCormick envelope method combined with SLSQP achieves a good balance between optimization effectiveness and computational efficiency. Its computation time is only 0.20 s, with a total profit of 19 151.72 yuan, demonstrating its practical advantages. The performances of SA, PSO, GA, and TS vary to some extent. The McCormick envelope method combined with spatial branch-and-bound takes 4.07 s, achieving the same total profit of 19 151.72 yuan, though with a slightly longer runtime. Among heuristic search algorithms, SA requires 12.52 s with a total profit of 19 151.72 yuan, which is relatively time-consuming. PSO delivers a poorer optimization outcome, with a total profit of 19 066.73 yuan and a runtime of 7.74 s. GA attains a relatively good result – total profit of 19 148.56 yuan – but has the longest runtime at 1644 s (about 27 min). TS has the shortest computation time at just 0.16 s, but the worst optimization effect, with a total profit of 18 967.33 yuan.

Overall, the McCormick envelope method combined with SLSQP offers solid computational efficiency and stable optimization performance, making it suitable for real-world scenarios with stringent computational time requirements.

5. Conclusion

This paper proposes a decision-making framework based on demand forecasting and optimization to address the optimization of procurement quantities and pricing strategies for perishable vegetables. In the demand forecasting stage, we employ a KAN-LSTM deep learning model. Through five-fold time-series cross-validation and multiple experiments, the model demonstrates strong predictive capability and robustness, with notable advantages in capturing time-series characteristics of sales and handling forecasting tasks across different time windows. The KAN-LSTM model achieves R^2 values above 0.90 across different datasets and exhibits low prediction errors in both sales forecasting and replenishment price forecasting, highlighting its superiority in dealing with complex time-series data.

In the optimization stage, we combine the McCormick envelope method with SLSQP to optimize inbound quantities and markups for the next 15 days, with the goal of maximizing total market profit. The optimization results show that this approach not only improves computational efficiency but also ensures the rationality and accuracy of the solutions. Compared with other global optimization methods (such as particle swarm optimization and genetic algorithms), the McCormick envelope method combined with SLSQP demonstrates higher computational efficiency and stable optimization performance in practical applications. The findings provide a scientific basis for procurement and pricing strategies in agricultural markets and open new avenues for applying demand forecasting and optimization algorithms in complex real-world scenarios.

Despite offering an effective framework for optimizing procurement and pricing strategies for perishable vegetables, this study still has some limitations. The model's predictive performance depends on the accuracy and completeness of historical data, while sales of perishable vegetables are influenced by seasonality and market fluctuations, factors that may not be fully captured in the data. Although the KAN-LSTM model performs well on time-series data, deep learning models typically require substantial data volume and computational resources, which may lead to performance bottlenecks when data are limited or computing power is constrained. The McCormick envelope method used in the optimization stage relies on certain assumptions about market conditions and cost parameters, which may vary in practice, potentially affecting the adaptability of the results. Future research could consider more dynamic market changes and a broader set of optimization algorithms to further enhance the model's applicability and accuracy.

Funding

This research supported by the National Undergraduate Innovation Training Program at the China University of Labor Relations, Project No. 124532025001 – Yuchi Li. This research was funded by the Undergraduate Research Program of the China University of Labor Relations, project number 25xjx030 – Yuchi Li. This research was also funded by the project Reform Practices of Classroom Teaching Models in the Context of Digital Intelligence, project number 2025JG41 – Xiali Li.

Data availability statement

Our data is open access at GitHub and cannot be used for commercial purposes without permission.

Author contribution statement

Conceptualization, Yuchi Li and Zhanli Li; methodology, Yuchi Li and Zhanli Li; software, Yuchi Li and Q.J.; validation, Yuchi Li and Q.J.; formal analysis, Yuchi Li and Jianhao Li; investigation, Yuchi Li and Q.J.; resources, Yuchi Li and Lili Xia; data curation, Lili Xia and Q.J.; writing-original draft preparation, Yuchi Li, Zhanli Li and Q.J.; writing-review and editing, Lili Xia; visualization, Lili Xia; supervision, Yuchi Li; project administration, Yuchi Li; funding acquisition, Lili Xia; Yuchi Li, Zhanli Li, Lili Xia and Q.J. are first contributors. All authors have read and agreed to the published version of the manuscript.

References

[1] B. Alhijawi and A. Awajan, Genetic algorithms: theory, genetic operators, solutions, and applications. *Evol. Intell.* **17** (2024) 1245–1256.

[2] E. Badakhshan, N. Mustafee and R. Bahadori, Application of simulation and machine learning in supply chain management: a synthesis of the literature using the SIM-ML literature classification framework. *Comput. Ind. Eng.* **198** (2024) 110649.

[3] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter and S. Hochreiter, XLSTM: extended long short-term memory. Preprint arXiv:2405.04517 (2024).

[4] Y. Belkourchia, L. Azrar and E.-S.M. Zeriab, A hybrid optimization algorithm for solving constrained engineering design problems, in 2019 5th International Conference on Optimization and Applications (ICOA) (2019) 1–7.

[5] A. Bell, M. Fairbrother and K. Jones, Fixed and random effects models: making an informed choice. *Qual. Quant.* **53** (2019) 1051–1074.

[6] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press (2004).

[7] J. Braun and M. Griebel, On a constructive proof of Kolmogorov's superposition theorem. *Constr. Approx.* **30** (20096) 653–675.

[8] H.W. Chen and A. Lim, A network price elasticity of demand model with product substitution. *J. Revenue Pricing Manag.* **22** (2023) 235–247.

[9] H.Y.S. Chien, J.S. Turek, N. Beckage, V.A. Vo, C. J. Honey and T.L. Willke, Slower is better: revisiting the forgetting mechanism in LSTM for slower information decay. Preprint arXiv:2105.05944 (2021).

[10] G. Ellison and S.F. Ellison, Search, obfuscation, and price elasticities on the internet. *Econometrica* **77** (2009) 427–452.

[11] Z. Fu, X. Yang, Y. Ma, Y. Sun and T. Wang, Integrating explainable AI and causal inference to unveil regional air quality drivers in China. *J. Environ. Manag.* **390** (2025) 126270.

[12] A.G. Gad, Particle swarm optimization algorithm and its applications: a systematic review. *Arch. Comput. Methods Eng.* **29** (2022) 2531–2561.

[13] R. Genet and H. Inzirillo, TKAN: temporal Kolmogorov–Arnold networks. Preprint arXiv:2405.07344 (2024).

[14] B. Gülmez, Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Syst. Appl.* **227** (2023) 120346.

[15] Z. Gong, Optimization of cross-border e-commerce (CBEC) supply chain management based on fuzzy logic and auction theory. *Sci. Rep.* **14** (2024) 14088.

[16] O. Hashemi-Amiri, F. Ghorbani and R. Ji, Integrated supplier selection, scheduling, and routing problem for perishable product supply chain: a distributionally robust approach. *Comput. Ind. Eng.* **175** (2023) 108845.

[17] S. Hochreiter and J. Schmidhuber, Long short-term memory. *Neural Comput.* **9** (1997) 1735–1780.

[18] E. Hofmann and E. Rutschmann, Big data analytics and demand forecasting in supply chains: a conceptual analysis. *Int. J. Logist. Manag.* **29** (2018) 739–766.

[19] V. Ismailov, On the representation by linear superpositions. *J. Approx. Theory* **151** (2008) 113–125.

[20] W.A. Kamakura, B.-D. Kim and J. Lee, Modeling preference and structural heterogeneity in consumer choice. *Mark. Sci.* **15** (1996) 152–172.

[21] J. Kennedy and R. Eberhart, Particle swarm optimization, in Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, WA, Australia (1995) 1942–1948.

[22] Y. Kong, Z. Wang, Y. Nie, T. Zhou, S. Zohren, Y. Liang, P. Sun and Q. Wen, Unlocking the power of LSTM for long term time series forecasting. *Proc. AAAI Conf. Artif. Intell.* **39** (2025) 11968–11976.

[23] C. Lea, R. Vidal, A. Reiter and G.D. Hager, Temporal convolutional networks: a unified approach to action segmentation, in European Conference on Computer Vision (ECCV), Vol. 9915 (2016) 47–54.

[24] Y. Li, Z. Song, L. Xia, J. Sun and Z. Wang, Application of the adaptive sparrow search algorithm in medical supply engineering. *Sci. Rep.* **15** (2025) 35775.

[25] Z. Li and Z. Yang, ESG rating disagreement and corporate total factor productivity: inference and prediction. *Finance Res. Lett.* **78** (2025) 107127.

[26] J. Lu, X. Zheng, E. Nervino, Y. Li, Z. Xu and Y. Xu, Retail store location screening: a machine learning-based approach. *J. Retail. Consum. Serv.* **77** (2024) 103620.

[27] A.S. Mahapatra, S. Sengupta, A. Dasgupta, B. Sarkar and R.T. Goswami, What is the impact of demand patterns on integrated online-offline and buy-online-pickup in-store (BOPS) retail in a smart supply chain management? *J. Retail. Consum. Serv.* **82** (2025) 104093.

[28] C. Mangla, M. Ahmad and M. Uddin, Optimization of complex nonlinear systems using genetic algorithm. *Int. J. Inf. Technol.* **13** (2021) 1913–1925.

[29] M.K. Mantrala, P.B. Seetharaman, R. Kaul, S. Gopalakrishna and A. Stam, Optimal pricing strategies for an automotive aftermarket retailer. *J. Mark. Res.* **43** (2006) 588–604.

[30] S. Mehroliya, S. Tomar, A. Arya and B. Singh, A novel hybrid GA-PSO algorithm-based optimization of transmission and expansion planning. *SN Comput. Sci.* **4** (2023) 690.

[31] F. Milisav, V. Bazinet, R.F. Betzel and B. Misic, A simulated annealing algorithm for randomizing weighted networks. *Nat. Comput. Sci.* **5** (2025) 48–64.
[32] D. Mircetic, B. Rostami-Tabar, S. Nikolicic and M. Maslaric, Forecasting hierarchical time series in supply chains: an empirical investigation. *Int. J. Prod. Res.* **60** (2021) 2514–2533.
[33] N. Niroumandrad, N. Lahrichi and A. Lodi, Learning tabu search algorithms: a scheduling application. *Comput. Oper. Res.* **170** (2024) 106751.
[34] J.J. Rushchitsky, On the constants of the nonlinear Murnaghan's hyperelastic material model. *Int. Appl. Mech.* **52** (2016) 508–519.
[35] R.S. Tavares, T.C. Martins and M.S.G. Tsuzuki, Simulated annealing with adaptive neighborhood: a case study in off-line robot path planning. *Expert Syst. Appl.* **38** (2011) 2951–2965.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in Advances in Neural Information Processing Systems (NeurIPS), Vol. 30 (2017) 5998–6008.
[37] R.G. Walters and W. Bommer, Measuring the impact of product and promotion-related factors on product category price elasticities. *J. Bus. Res.* **36** (1996) 203–216.
[38] A. Wan, Q. Chang, K. AL-Bukhaiti and J. He, Short-term power load forecasting for combined heat and power using CNN-LSTM enhanced by attention mechanism. *Energy* **282** (2023) 128274.
[39] L. Yunzhu, Improvement of simulated annealing algorithm and its application in water environment engineering, in Frontier Computing, FC 2022, edited by J.C. Hung, N.Y. Yen, and J.W. Chang, Vol. 1031 of *Lecture Notes in Electrical Engineering*. Springer, Singapore (2023) 1717–1729.
[40] O. Zabraoui, Y. Hmamou, A. Chafi and S. Kammouri Alami, A comparative study of multi-algorithm optimization for inventory analytics in supply chains. *Supply Chain Anal.* **12** (2025) 100154.
[41] J. Zhang, H. Jiang, X. Li, L. Zhang and Z. Li, A CNN-transformer hybrid approach for decoding visual neural activities evoked by natural images into texts. *NeuroImage* **247** (2022) 118377.

Appendix A.

The Kolmogorov–Arnold Representation Theorem states that any multivariate continuous function $f(x_1, x_2, \ldots, x_n)$ can be represented as a sum and composition of univariate functions [7], namely:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{A.1}$$

where $\phi_{q,p} : [0,1] \to \mathbb{R}$ are univariate mapping functions, and $\Phi_q : \mathbb{R} \to \mathbb{R}$ are another set of univariate mapping functions.

The KAN network architecture is shown in Figure A.1. Leveraging this theorem, KAN learns activation functions at the connections of the neural network, rather than using traditional fixed activations (such as ReLU or Sigmoid). In a KAN network, the computation at each layer is given by:

$$x_{l+1,j} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}). \tag{A.2}$$

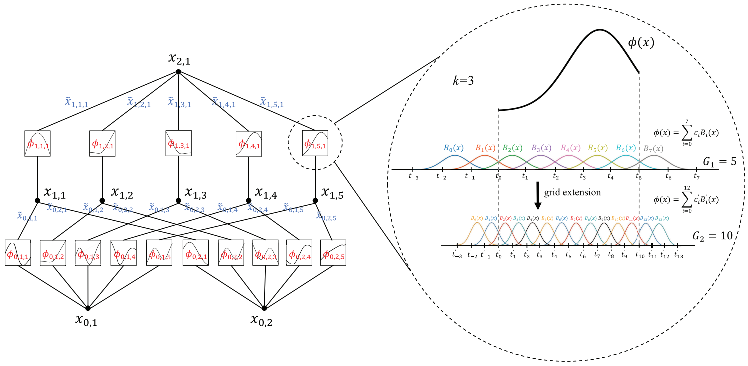


FIGURE A.1. KAN working framework.

That is:

$$x_{l+1} = \Phi_l x_l \tag{A.3}$$

where Φ_l denotes the function matrix of the KAN network. The overall computation of a KAN network can be expressed as:

$$\mathrm{KAN}(x) = (\Phi_L \circ \Phi_{L-1} \circ \cdots \circ \Phi_1 \circ \Phi_0)x. \tag{A.4}$$

This construction endows KAN with stronger nonlinear modeling capacity and superior function-approximation properties.

APPENDIX B

This appendix analyzes the curvature properties of the proposed optimization model under a simplified setting and a general interdependent setting. In the simplified case, analytical second-derivative conditions can be derived for the one-dimensional markup term. In the general case, however, price and demand interdependencies introduce coupled nonlinear terms, making a direct proof of global convexity challenging. Therefore, the numerical Hessian analysis reported below is used only to illustrate local curvature features of the objective landscape, while McCormick envelope relaxations are adopted in the computational phase to handle the nonlinear terms.

To facilitate the analysis, we use the following lemmas.

Lemma 1. *Preservation of curvature under positive weighted linear combinations. If $f_i(x)$ are convex functions, then their positive weighted linear combination $\sum_i a_i f_i(x)$, where $a_i > 0$, is also convex. Likewise, if $f_i(x)$ are concave functions, then $\sum_i a_i f_i(x)$, where $a_i > 0$, is also concave.*

Lemma 2. *Curvature preservation under subtraction of a linear function. Given a function $g(x)$ and a linear function $h(x) = ax + b$, if $g(x)$ is concave, then $f(x) = g(x) - h(x)$ is also concave; if $g(x)$ is convex, then $f(x) = g(x) - h(x)$ is also convex [6].*

The objective function is:

$$\max_{Q,m} \text{Profit} = \sum_i [(p_i - c_i)D_i - h_i(Q_i - D_i)]. \tag{A.5}$$

Since only Q_i and m_i are decision variables in the objective, it can be rewritten as:

$$\begin{aligned}
\max_{Q,m} \text{Profit} &= \sum_i [(p_i - c_i)D_i - h_i(Q_i - D_i)] \\
&= \sum_i [(c_i + c_i m_i - c_i)D_i - h_i Q_i + h_i D_i] \\
&= \sum_i [c_i m_i D_i - h_i Q_i + h_i D_i] \\
&= \sum_i (c_i m_i + h_i)D_i - \sum_i h_i Q_i.
\end{aligned} \tag{A.6}$$

According to Lemmas 1 and 2, the curvature of the objective depends on the term $(c_i m_i + h_i) D_i$. Let

$$f_i(m_i) = (c_i m_i + h_i) D_i. \tag{A.7}$$

Case 1: Simplified setting
Assume $\gamma_j = \delta_j = 0$, *i.e.*, the recursive terms $p_j^{\gamma_j}$ and $D_j^{\delta_j}$ are omitted. Then:

$$D_i = e^{\alpha} p_i^{-\beta_i}, \tag{A.8}$$

$$f_i(m_i) = (c_i m_i + h_i) e^{\alpha} p_i^{-\beta_i}. \tag{A.9}$$

Since $p_i = c_i(1 + m_i)$, we have:

$$\frac{dp_i}{dm_i} = c_i. \tag{A.10}$$

Let $u(m_i) = c_i m_i + h_i$ and $v(m_i) = p_i^{-\beta_i}$. Then

$$f_i(m_i) = e^{\alpha} u(m_i) v(m_i). \tag{A.11}$$

Using the product rule,

$$f_i'(m_i) = e^{\alpha} \left[u'(m_i) v(m_i) + u(m_i) v'(m_i) \right]. \tag{A.12}$$

For the two parts,

$$u'(m_i) = c_i, \tag{A.13}$$

$$v'(m_i) = -\beta_i c_i^{-\beta_i} (1 + m_i)^{-\beta_i - 1}. \tag{A.14}$$

Substituting back gives

$$\begin{aligned} f_i'(m_i) &= e^{\alpha} \left[c_i^{-\beta_i + 1} (1 + m_i)^{-\beta_i} + (c_i m_i + h_i)(-\beta_i) c_i^{-\beta_i} (1 + m_i)^{-\beta_i - 1} \right] \\ &= \frac{e^{\alpha}}{c_i^{\beta_i} (1 + m_i)^{\beta_i + 1}} [c_i(1 + m_i) - \beta_i (c_i m_i + h_i)]. \end{aligned} \tag{A.15}$$

Define $g(m_i) = (1 + m_i)^{-\beta_i - 1}$ and $W(m_i) = c_i(1 + m_i) - \beta_i(c_i m_i + h_i)$. Then

$$f_i'(m_i) = \frac{e^{\alpha}}{c_i^{\beta_i}} g(m_i) W(m_i). \tag{A.16}$$

Differentiating again,

$$f_i''(m_i) = \frac{e^{\alpha}}{c_i^{\beta_i}} \left[g'(m_i) W(m_i) + g(m_i) W'(m_i) \right]. \tag{A.17}$$

The derivatives are:

$$g'(m_i) = -(\beta_i + 1)(1 + m_i)^{-\beta_i - 2}, \tag{A.18}$$

$$W'(m_i) = c_i(1 - \beta_i). \tag{A.19}$$

Substituting into the second derivative yields

$$\begin{aligned} f_i''(m_i) &= \frac{e^{\alpha}}{c_i^{\beta_i} (1 + m_i)^{\beta_i + 2}} [-2\beta_i c_i (1 + m_i) + (\beta_i + 1)\beta_i (c_i m_i + h_i)] \\ &= \frac{e^{\alpha} \beta_i}{c_i^{\beta_i} (1 + m_i)^{\beta_i + 2}} [-2c_i(1 + m_i) + (\beta_i + 1)(c_i m_i + h_i)] \\ &= \frac{-e^{\alpha} \beta_i c_i}{c_i^{\beta_i - 1} (1 + m_i)^{\beta_i + 2}} \left[(1 - \beta_i) m_i - (1 + \beta_i) \frac{h_i}{c_i} + 2 \right]. \end{aligned} \tag{A.20}$$

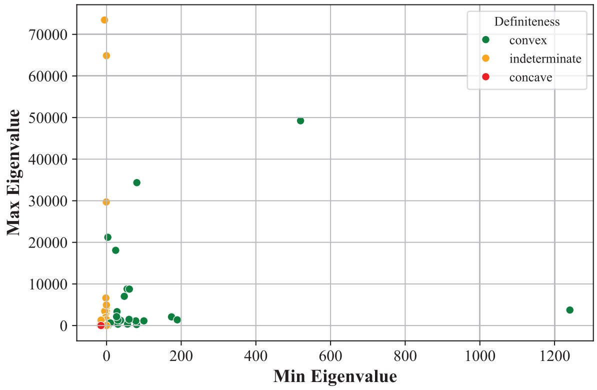


FIGURE A.2. Eigenvalues of the Hessian matrix.

Since $e^\alpha > 0$, $\beta_i > 0$, $c_i^{\beta_i - 1} > 0$, and $(1+m_i)^{\beta_i+2} > 0$, the sign of the second derivative is determined by

$$S(m_i) = (1-\beta_i)m_i - (1+\beta_i)\frac{h_i}{c_i} + 2. \tag{A.21}$$

Therefore, the sign pattern of $S(m_i)$ characterizes the local curvature of the simplified one-dimensional objective term. In particular, when $S(m_i) > 0$, we have $f_i''(m_i) < 0$, indicating local concavity of the simplified term; when $S(m_i) < 0$, the term is locally convex. This result provides analytical insight into how the curvature depends on the elasticity parameter β_i, the markup decision m_i, and the spoilage-cost ratio h_i/c_i. However, this analysis applies only to the simplified setting and should not be interpreted as a proof of global convexity for the full interdependent optimization model.

Case 2: General interdependent setting

Assume $\gamma_j \neq 0$ and $\delta_j \neq 0$, so that the recursive terms $p_j^{\gamma_j}$ and $D_j^{\delta_j}$ are retained. Then:

$$D_i = e^\alpha p_i^{-\beta_i} \prod_j p_j^{\gamma_j} \prod_j D_j^{\delta_j}. \tag{A.22}$$

In this general setting, price and demand interdependencies introduce coupled nonlinear terms into the objective, substantially increasing the model complexity. As a result, a direct proof of global convexity is challenging. To illustrate the local structure of the objective landscape, we employ numerical Hessian eigenvalue analysis. Specifically, the Hessian matrix is defined as

$$H_f(m_i) = \begin{pmatrix} \frac{\partial^2 f}{\partial m_1^2} & \frac{\partial^2 f}{\partial m_1 \partial m_2} & \cdots & \frac{\partial^2 f}{\partial m_1 \partial m_n} \\ \frac{\partial^2 f}{\partial m_2 \partial m_1} & \frac{\partial^2 f}{\partial m_2^2} & \cdots & \frac{\partial^2 f}{\partial m_2 \partial m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial m_n \partial m_1} & \frac{\partial^2 f}{\partial m_n \partial m_2} & \cdots & \frac{\partial^2 f}{\partial m_n^2} \end{pmatrix}. \tag{A.23}$$

Here, f denotes the objective function, $m_i = (m_1, m_2, \ldots, m_n)$ is the decision-variable vector, and $H_f(m_i)$ is the Hessian evaluated at that point. The eigenvalue results of the Hessian matrices computed at 100 randomly sampled decision-variable vectors are shown in Figure A.2.

As shown in Figure A.2, the minimum and maximum eigenvalues vary across sampled points, and many regions exhibit indefinite Hessians. This indicates that the objective may display locally convex, locally concave, or saddle-type behavior in different parts of the decision space. Therefore, the Hessian analysis is used only as a local curvature diagnostic to illustrate the structural complexity of the general problem; it does not establish global convexity of the original optimization model.

Given this potential non-convexity, the computational strategy adopted in this study is based on McCormick envelope relaxations. Specifically, auxiliary variables and envelope constraints are introduced for the nonlinear interaction terms to construct convex relaxations, thereby transforming the original problem into a form amenable to gradient-based solvers such as SLSQP. This relaxation-based treatment improves tractability and numerical stability for the general interdependent setting.

Appendix C

The hyperparameters of Simulated Annealing (SA) [35], Particle Swarm Optimization (PSO) [21], Genetic Algorithm (GA) [1], and Tabu Search (TS) [33] are shown in Table A.1:

Table A.1. Hyperparameter settings.

Algorithm	Parameter	Default value
Tabu Search	Maximum Iterations	100
	Tabu List Tenure	5
	Neighborhood Size	5
	Step Range	−0.1 to 0.1
	Initial Solution (Quantity)	10% of sales volume
	Initial Solution (Markup)	0.1
Simulated Annealing	Maximum Iterations	1000
	Initial Temperature	1.0
	Minimum Temperature	0.0001
	Step Size	0.1
	Neighborhood Size	10
Particle Swarm Optimization	Swarm Size	100
	Maximum Iterations	1000
	Initial Solution	10% of sales volume
Genetic Algorithm	Swarm Size	100
	Maximum Iterations	1000
	Mutation Probability	0.1
	Crossover Probability	0.8