University of Tehran Press

Industrial Management Journal

Online ISSN: 3115 - 7386

Homepage: https://imj.ut.ac.ir

Integrating Stock Levels into Demand Forecasting and Discount Optimization for Retail

Arezou Mansouri ✉[1], **Mansour Momeni** [2], and **Mina Kazemi Miyangaskary**[3]

1. Corresponding author, Ph.D. Candidate, Department of Technology and Innovation Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. E-mail: arezoumansouri@ut.ac.ir
2. Prof., Department of Technology and Innovation Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. E-mail: mmomeni@ut.ac.ir
3. Postdoctoral Researcher, Department of Management, University of Québec at Trois-Rivières, QC, Canada. E-mail: mina.kazemi.miyangaskary@uqtr.ca

Article Info

Article type:
Research Article

Article history:
Received October 02, 2025
Received in revised form November 06, 2025
Accepted January 24, 2026
Available online February 20, 2026

Keywords:
Machine learning; deep neural network; parameter tuning; stock level features; discount ladder.

ABSTRACT

Objective: Accurate sales forecasting is crucial for inventory management and supply chain planning in retail settings with high product variety, short life cycles, and volatile demand. Errors cause excess stock, stock-outs, lost sales, suboptimal promotions, higher costs, reduced customer satisfaction, and brand damage. Reliable forecasts drive replenishment, pricing, promotion design, inventory allocation, and profitability. As retailers embrace data-driven strategies, models must account for interactions among demand, pricing, and inventory. This study introduces a deep neural network framework that jointly tackles demand forecasting and discount optimization.

Methodology: Unlike traditional approaches that ignore inventory limits, the model uses store-level stock availability as an input. This captures reality helping separate true demand from inventory-shortage effects. Experiments show this inventory-aware method substantially outperforms baselines without stock data on standard accuracy metrics. The framework also features a heuristic ladder search algorithm for discount optimization. It uses deep learning forecasts to evaluate discrete discount options, balancing demand uplift, remaining inventory, and profit margins. This prevents excessive markdowns that hurt margins or weak discounts that leave excess stock.

Results: Tested on real-world data across products and stores, the approach yields better forecasts and notable profit gains by aligning pricing with inventory and demand.

Conclusion: Overall, integrating demand forecasting and discount optimization outperforms separate handling, delivering retailers a practical tool to enhance inventory efficiency, promotion effectiveness, and profitability.

Cite this article: Mansouri, A., Momeni, M., & Kazemi Miyangaskary, M., (2026). Integrating stock levels into demand forecasting and discount optimization for retail, *Industrial Management Journal*, 18(2), 183-207. https://doi.org/10.22059/imj.2026.408958.1008279

Publisher: University of Tehran Press.

DOI: https://doi.org/10.22059/imj.2026.408958.1008279

Introduction

Advances in machine learning and artificial intelligence have created substantial opportunities for enhancing supply chain performance. At the 2018 MIT Center for Transportation and Logistics conference on machine learning and artificial intelligence, researchers highlighted the growing influence of data-driven methods across several supply chain domains, including demand forecasting, pricing, discount optimization, and logistics (Cabbalero & Rice, 2018). Their findings, consistent with much of the recent literature, underscore the ability of AI and ML based models to improve supply chain efficiency through more accurate predictions and optimized decision-making.

In supply chains, efficiency improvements typically involve reducing stockouts, avoiding excess inventory, lowering ordering and holding costs, and improving the reliability of goods and service delivery. To support such improvements, a significant body of research has examined how analytical and learning-based techniques can help answer essential managerial questions, how much inventory should be ordered and how can firms minimize shortages, control inventory levels, reduce ordering costs, and ultimately maximize profits.

Ghamete et al. (2021) emphasize that a central challenge in supply chain management is the estimation of stochastic parameters. Inaccurate estimates of future demand, consumer preference shifts, supplier lead times, or production downtimes can result in inefficiencies, forcing firms, particularly retailers, to overinvest in capacity or stock, tying up excessive capital. For example, overestimating warehouse capacity requirements due to volatile demand can lead to unnecessary infrastructure investment.

To address uncertainty in nondeterministic parameters, the literature generally describes two strategies. The first is the use of safety factors to buffer against shortages, while the second seeks to optimize capital allocation. According to Reid and Sanders (2019), safety factor approaches often lead to overestimation of capacity and are less effective in supply chains dealing with perishable goods, whereas capital-optimization approaches may tolerate lost demand and are more suitable in environments with backorders or limited competition.

Accurate estimation of uncertain parameters is central to improving supply chain performance. Conventional retail forecasting models typically assume that observed sales equal realized demand, thereby abstracting from inventory constraints. However, when stockouts occur, sales represent a censored realization of latent demand, leading to systematic downward bias in demand estimation. This issue has been well documented in the operations literature on lost sales and demand censoring, which shows that failure to account for inventory availability distorts both statistical inference and operational decisions. In particular, censored demand induces biased replenishment parameters, as under-observed demand translates into systematically lower forecasts and, consequently, persistent under-ordering. Moreover, stockouts contaminate managerial learning by masking lost sales as demand decline, thereby affecting assortment planning and product evaluation. When promotions are run under binding inventory constraints, estimated lift and price elasticity are attenuated, leading to promotion miscalibration and suboptimal pricing decisions. Thus, treating demand as independent of inventory does not merely increase operational costs; it structurally distorts the feedback loop between demand learning and inventory control (Ananth Raman & Marshall Fisher, 2001; Neslin et al., 2006; Vishal Gaur et al., 2007).

Motivated by this need, the present research develops a demand forecasting model tailored to the retail sector, explicitly examining the role of stock levels as a predictive feature. Furthermore, we propose a discount optimization algorithm that uses the resulting demand function to compute optimal price reductions.

The remainder of this paper is organized as follows:

Section 2 reviews the state-of-art and related literature. Section 3 addresses research methodology. Section 4 focuses on deep learning method to explain its beneficial aspects which motivated the authors to employ it. Section 5 introduces the proposed demand forecasting model and describes the training and parameter-tuning procedures. Section 6 presents the experimental results and analyzes model performance. Section 7 concludes the paper with key findings and contributions.

Literature Background

Demand forecasting is a central challenge in retail operations and serves as a critical input for a wide range of managerial decisions, including inventory planning, replenishment scheduling, and promotional design (Huber & Stuckenschmidt, 2020; Kazemian et al., 2023; Rezasoltani, 2024). The complexity of forecasting increases substantially in retail environments characterized by large product assortments and heterogeneous demand patterns. Moreover, the high dimensionality of relevant features, such as seasonality, promotional effects, and temporal dependencies, introduces substantial computational difficulties that further complicate the development of accurate forecasting models (TaghizadehYazdi, et al. 2015).

Beyond operational planning, demand forecasts play a pivotal role in pricing and discount decisions, as expected demand responses directly influence the effectiveness and profitability of promotional strategies. Inaccurate demand estimates may therefore not only distort inventory decisions but also lead to suboptimal discount levels that either fail to stimulate demand or unnecessarily erode profit margins.

A significant body of supply chain research emphasizes the importance of accurately estimating stochastic parameters such as demand, lead times, and consumer behavioral shifts. Ghamete et al. (2021) have argued that insufficient estimation of these uncertain parameters often leads firms to overinvest in inventory or capacity, thereby increasing the level of capital tied up in the system. For instance, demand overestimation may cause retailers to allocate excessive warehouse capacity or maintain unnecessarily high stock levels, which can adversely affect cost efficiency.

Two general methodological approaches have been proposed in the literature to address variability in nondeterministic parameters. The first approach introduces safety factors to mitigate the risk of shortages or capacity constraints, while the second focuses on optimizing capital usage under uncertainty. Reid and Sanders (2019) noted that safety-factor–based strategies frequently lead to conservative planning and overestimated capacity, which limits their applicability in settings involving perishable goods. In contrast, capital-optimization approaches, although more resource-efficient, may lead to unmet demand and are therefore most suitable for environments where backordering is feasible or competitive intensity is comparatively low.

Given these limitations, the development of more accurate and robust forecasting techniques remains essential. A growing stream of research has consequently explored the use of machine learning and hybrid modeling strategies to enhance predictive performance. Baradwaj and Gunasekaran (2025) proposed a hybrid regression framework that aggregates predictions from multiple regression models and demonstrate that this approach achieves 84.61% accuracy, outperforming ARIMA by roughly 14% in terms of MAPE. Building on multi-factor forecasting frameworks, Siam et al. (2024) introduced a four-stage demand forecasting model that incorporates weather-related variables, promotional activities, historical sales patterns, and calendar effects, employing Gradient Boosting Machines (GBM) as the primary learning algorithm.

Research has also examined approaches suited to sparse or limited data environments. For products with insufficient historical observations, Nagai et al. (2024) proposed an ARIMA-based zero-shot learning method that leverages structural information rather than long-term sales histories. In addition, deep learning architectures have gained increasing attention for their ability to model nonlinear temporal dependencies and cross-product interactions. Aktas et al. (2024) developed a graph-based LSTM model (LSTMgraph) that captures demand correlations across products using three types of relationships: weekly demand co-movement, co-occurrence in customer shopping baskets, and transaction-level similarities. Further advancements in deep-learning–based retail forecasting can be found in the studies of Yang et al. (2025) and Liu et al. (2025), which explore advanced representation-learning techniques to improve predictive accuracy.

Joseph et a. (2022) proposed a hybrid deep learning framework to forecast sales. The framework consists of two deep networks, say convolutional neural network to forecast initial values of demand and in sequence with it a LSTM network to get inventory score and provide final results.

Because of the beneficial aspects of demand forecasting, many researchers have put forward to employ the outcomes of forecasting in pricing and promotion optimization (Horden & Bisset, 2022).

Sütçü and Yıldız (2025) proposed a stochastic framework to optimize prices in retail section based on a demand forecasting model. They analyzed the relation of price and demand and concluded that the behavior is s-shaped. When the price decreases, the demand increases exponentially. A stochastic dual dynamic programming to approximate optimal pricing decisions.

One of the most novel approaches for both demand forecasting and discount optimization is machine learning (ML). ML is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from data and make predictions or decisions without these being explicitly programmed. The fundamental idea behind ML is to allow systems to automatically improve their performance over time by learning from experience. A wide variety of algorithms exist in the field of ML. Broadly speaking, these techniques fall into supervised, and unsupervised learning techniques. In supervised learning, the algorithm learns from a labeled dataset (such as historic demand), whereas unsupervised techniques find patterns in data that are not labeled. Within these categories, there are a multitude of algorithms such as Artificial Neural Networks (ANNs), linear regression, decision trees, and others. In this regard, one can refer to the review paper conducted by Bergsma et al. (2025) to dig out details of different algorithms of ML used to forecast demand and post-employment of the obtained demand function.

Meisheri et al. (2022) designed a reinforcement learning algorithm to forecast the demand and optimize discounts for pricing purposes. They revealed that using deep learning can lead to accurate results because of its flexibility governed by the parameters and activation function. However, their study does not consider stock level of goods in the stores. Addressing this gap, the present study proposes a deep neural network–based demand forecasting framework that incorporates store-level stock availability as a key explanatory feature. The innovation of this research lies in the integrated treatment of demand forecasting and discount optimization within a unified, inventory-aware framework. Unlike conventional retail forecasting models that assume observed sales fully reflect true demand, the proposed approach explicitly incorporates store-level stock availability as an input to a deep neural network, allowing the model to distinguish between demand fluctuations and inventory-induced sales constraints. Additionally, the study introduces a practical discount ladder optimization mechanism that directly leverages forecasted demand to guide pricing decisions under inventory limitations. By jointly optimizing forecasting accuracy and pricing effectiveness, the proposed framework bridges the gap between predictive analytics and operational decision-making in retail, offering a scalable and data-driven solution that enhances both forecasting reliability and profitability. Together, these contributions aim to enhance the accuracy and decision-making relevance of forecasting models in retail supply chains.

Materials and Methods

This study employs a quantitative, data-driven methodology to develop an integrated framework for retail demand forecasting and discount optimization. Real-world retail data containing sales, pricing, calendar variables, and store-level stock availability are preprocessed and used for model development. A stock level score is engineered as an additional input feature to capture inventory constraints and distinguish true demand from stock-limited sales. Demand forecasting is performed using a deep neural network, with key architectural parameters optimized through the Particle Swarm Optimization algorithm. Forecast accuracy is evaluated using the Mean Absolute Error metric. Based on the forecasted demand, a heuristic discount ladder search algorithm is applied to identify discount levels that maximize total profit while accounting for inventory constraints. The framework is evaluated in terms of both forecasting accuracy and profitability to demonstrate the effectiveness of the proposed integrated approach.

Deep Learning

Early forecasting methods are statistical in nature and hey rely on uncovering the statistical properties of historical data to make future predictions. Despite their simplicity, they are still used in a wide range of applications where the data is scarce and the underlying patterns are simple and easy to capture. With the rise of machine learning and big data, modelling more complex patterns and accounting for a multitude of factors affecting demand has become possible. However, this advantage remains unexploited for traditional univariate ML methods that model each time series in isolation and learn a separate model thereof. In contrast, multivariate models can learn from multiple time series simultaneously and integrate this information with exogenous predictors to produce accurate future forecasts. This can generally be achieved through global training strategies that allow information sharing across time series to learn the parameters of one global model (Wen et al., 2017).

Very close to the area of this research, say, retail sector, Riachy et al. (2025) revealed the power of deep learning in demand forecasting. They explained how most of machine learning techniques cannot be trained efficiently in the presence of data gap. The ability of deep learning methods to capitalise on the large amount of data that became available and effectively learn global models has positioned them at the forefront of leading demand forecasting algorithms (Peláez-Rodríguez et al., 2024).

Among the deep learning methods that were the first to gain popularity for time-series forecasting are Recurrent Neural Networks (RNNs) and in particular, Long Short-Term Memory (LSTM) networks (Mughees et al., 2021). This is due to their inherent ability to handle sequential data and retain temporal information. For instance, Fischer and Krauss (2018) apply an LSTM architecture to perform large-scale predictions for the financial markets. Li et al. (2017) proposed a Diffusion Convolutional Recurrent Neural Network (DCRNN) to forecast traffic on a road network. The model captures both spatial and temporal dependencies using bidirectional random walks and an encoder–decoder architecture. Bandara et al. (2020) employ an LSTM network preceded by a clustering step and other pre-processing techniques to enhance the forecasting accuracy on the CIF2016 dataset that contains data originating from the banking domain in addition to synthetic data. Mughees et al. (2021) design a bidirectional LSTM sequence-to-sequence regression approach to predict day-ahead peak electricity demand in Pakistan. It is also worth mentioning that RNNs have been extensively combined with traditional forecasting methods, such as ARIMA, to form hybrid models (Dave et al., 2021, Ji et al., 2019).

Proposed model

In this section, detailed framework of the model and solution approach will be discussed including the proposed features and dealing with network architecture.

1- Stock level score

In the demand forecasting, stock level plays a key role since it is one of the important features affect the result of model. Riazi et al. (2022) emphasizes the key role of shelf space in inventory control and demand forecasting of perishable products. Inspiring this effect, they proposed an inventory control scheme to avoid items expiration. They showed that increasing shelf space changes the demand and can reduce expiration rate. The forecasting algorithm is Holt-winters seasonal model.

Wu et al. (2024) discussed about shelf space and its effectiveness on category management. In this study, it has been shown that participating stock level can enhance the performance of forecasting model remarkably. Trapero et al. (2024) state that the demand behavior relative to stock level is a nonlinear and strictly increasing behavior where, as inventory increases, the demand growth rate tends toward zero. The selection of logarithmic fuzzification is specifically intended to mirror the diminishing marginal utility of inventory; this approach ensures high sensitivity and a steeper gradient at low stock levels where the risk of stockouts is most critical, whereas a sigmoid function's initial 'lag' would under-represent this urgency.

Accordingly, they described the relationship between these two variables using a logarithmic function. Figure 2 illustrates this pattern.

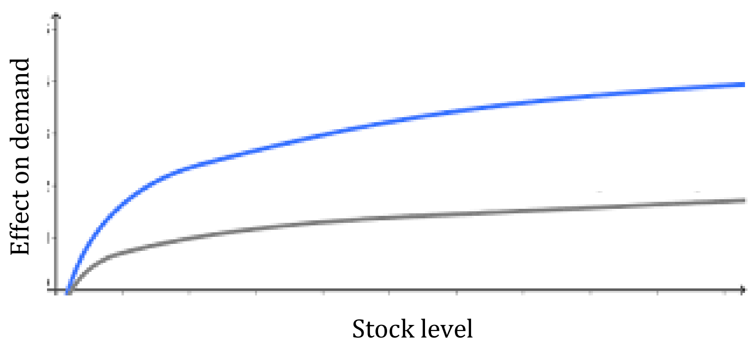

Figure 1. Effect of stock level on demand (Trapero et al., 2024)

Here, two approaches have been employed to model this feature, say, stock level score we show it with $\boldsymbol{\tau}$ letter. To describe more, assume that a chain of retail stores provides $\boldsymbol{P}$ products in $\boldsymbol{K}$ stores. Accordingly, let $\boldsymbol{s_p^k}, \{\boldsymbol{p = 1, \dots, P; k = 1, \dots, K}\}$ stands for inventory quantity of product $\boldsymbol{p}$ at store $\boldsymbol{k}$. Then stock level score is calculated by one of the two approaches at below.

(a) Binning: In this method, a discrete number is assigned as the stock level score of each store per product and finally, the average of the scores will be considered as score of that product. relation 1, shows the formula.

$$\tau_p = \frac{1 \upsilon_p}{K} \tag{1}$$

Where, υ_p is a vector of size K in which its kth element is the score bin of product p at store k and can be calculated by formula (1). In addition, $1_{1\times K}$ is vector of ones of size K.

$$\upsilon_p = min\left(\frac{s_p^k}{5}, 5\right), \forall k = 1, \dots K \tag{2}$$

(b) Fuzzy mapping function: In this method, a logarithmic function will be responsible for fuzzification and calculating the feature, similarly to the research done by Krackowitz et al (2009). Here, τ_p can be calculated same as formula 4, however, calculation of parameter υ_p is as follows:

$$\upsilon_p = log\left(10 \times \frac{(s_p^k+1)}{\max_{k=1,\dots,K}(s_p^k+1)}\right), \forall k = 1, \dots K \tag{3}$$

In fact, the above relation maps the final score as a continuous number in the range of 0 to 1 while maintaining a pattern similar to Figure 1.

2- Network topology and train algorithm

Deep neural network is a type of feedforward neural network that typically has a very high number of hidden layers and neurons in each layer. Figure 1 shows a common example of a deep neural network. The training process of this model is based on gradient descent algorithms, and the Adam algorithm is often used for training and weight optimization. Figure 2 shows the corresponding pseudocode (Kingma & Ba, 2014).

In addition, another important parameter is the activation functions related to each neuron. Due to the large number of neurons and connections between successive layers, determining the optimal activation functions and training the model becomes very difficult.

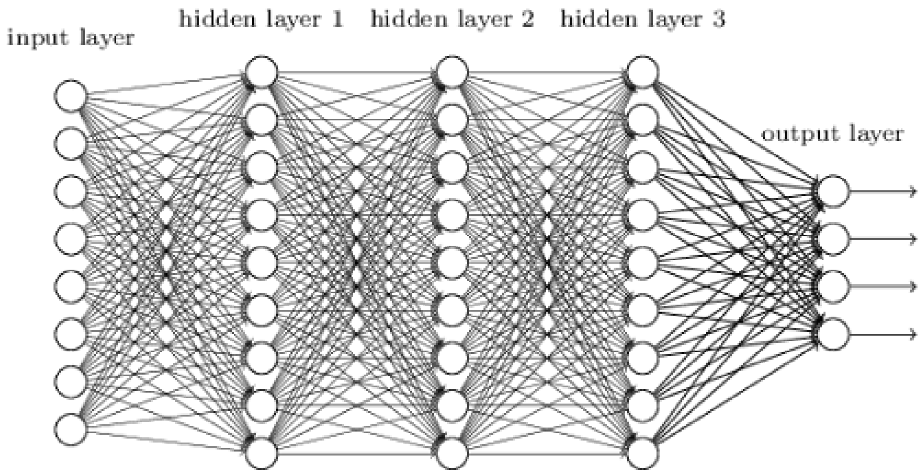

Figure 2. Deep Neural Network

To train the neural network used in this article, assume that the number of hidden layers and the number of neurons in each layer are represented as a pair $\{\boldsymbol{\lambda}, \boldsymbol{v}\}$, where $\boldsymbol{\lambda}$ is an integer indicating the number of hidden layers and $\boldsymbol{v}$ is an asymmetric list of length $\boldsymbol{\lambda}$ that determines the structure and number of neurons in each layer. More explicitly, $\boldsymbol{v_t}, \boldsymbol{t} = \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{\lambda}$ is the $\boldsymbol{t^{th}}$ component of the list $\boldsymbol{v}$, which is a vector of length $\boldsymbol{l^t}$. Here, $\boldsymbol{l^t}$ specifies the number of neurons in the $\boldsymbol{t^{th}}$ layer, and each component's value indicates the activation function type of that neuron, which will be one of the following: linear, Sigmoid, or $\boldsymbol{ReLU}$. The code for each activation function can be identified based on Table 2.

(1) Initialize weights
(2) For each epoch:
    Forward Propagation: Compute predictions
    Compute Loss
    Backpropagation: Compute gradients
(3) Update Parameters
(4) Repeat until convergence

Figure 3. Sudo-code of training deep network

Table 1. Coding of activation functions

Code		Activation function
1		Linear
2		Sigmoid
3		swish

The operational mechanism works as follows: In each iteration, the algorithm determines the number of hidden layers, the number of neurons in each layer, and the types of activation functions, then trains the network using the pseudocode shown in Figure 3. Performance metrics are stored, and in each iteration, the algorithm attempts to improve these metrics by modifying the number of hidden layers, the number of neurons in each layer, and the types of activation functions. These modifications and optimizations follow the Particle Swarm Optimization algorithm. Schematically, the corresponding flowchart is shown in Figure 4.

To maintain computational efficiency, the PSO explores a multi-output architecture where the hidden layers are shared across the SKU portfolio. This global search strategy ensures that the overhead of the evolutionary optimization is independent of the number of SKUs, effectively treating the entire forecasting task as a single, unified optimization objective.

3- Discount optimization

The discount optimizer behind the proposed framework is a search algorithm which mimics response surface methodology concepts. Generally speaking, after training the neural network a heuristic algorithm starts to change the value of discounts and get the value of sales based on the changed discount. At this step, the profit of sales is calculated. This procedure iterates to find optimal or near optimal discounts to maximize the profit.

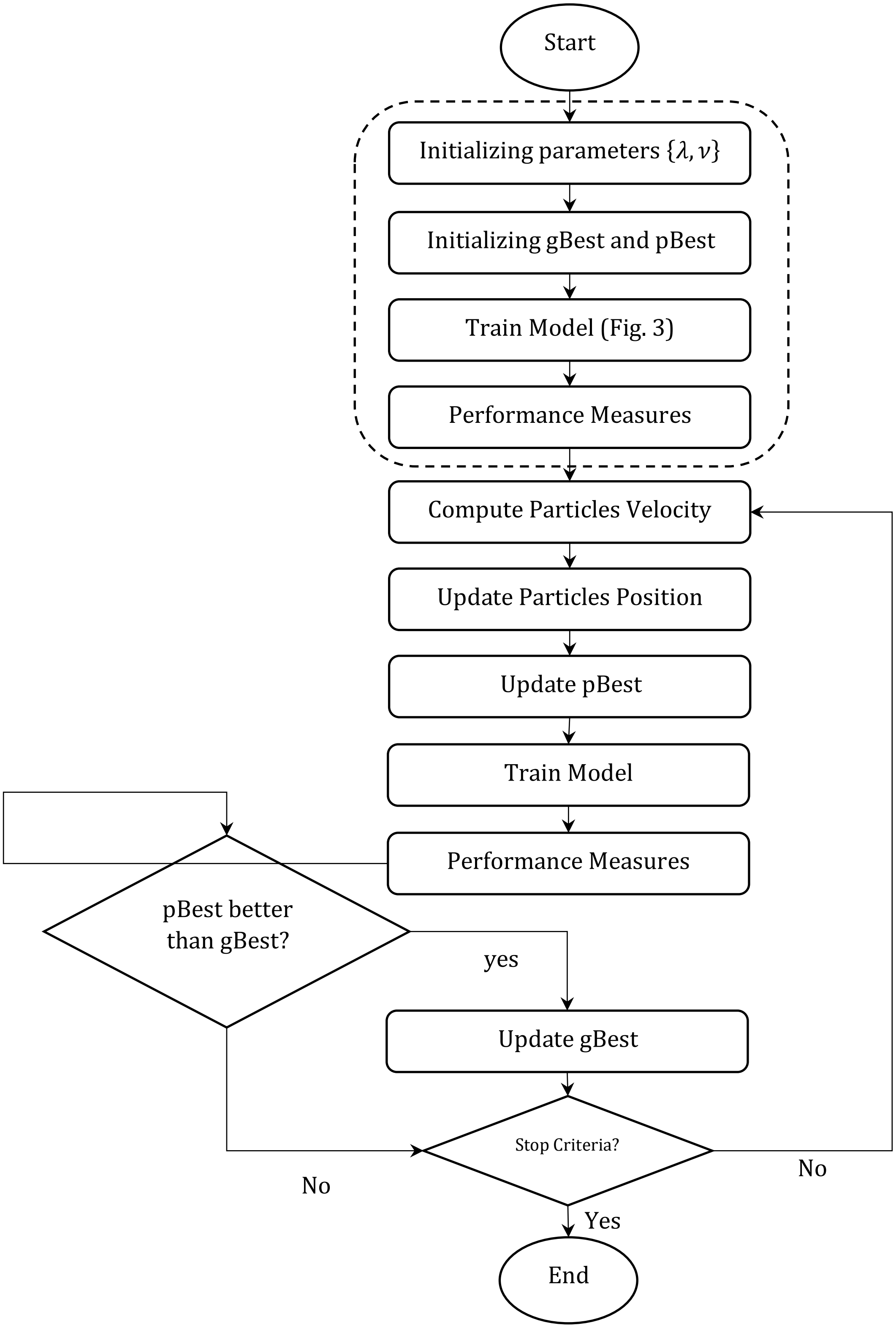

Figure 4. PSO algorithm for parameter tuning and model training

To facilitate the searching algorithm, a discount ladder is designed for each SKU with a fixed step size of 0.5 percent. This ladder defines a discrete set of feasible discount levels that the optimization algorithm can systematically explore. As illustrated in Figure 5, the algorithm moves along the discount ladder by incrementally adjusting the discount level of each SKU and, at each step, passes the selected discount value to the trained neural network to generate the corresponding demand forecast. Based on the predicted demand and available inventory, the resulting profit is then calculated, allowing the algorithm to identify the discount configuration that maximizes total profit.

Tang et al. (2004) studied the relation of discount percent and profit. According to their research, there is n optimal discount percent which leads to maximizing profit and the profit will reduced for larger and smaller values than the optimal discount (Figure 6).

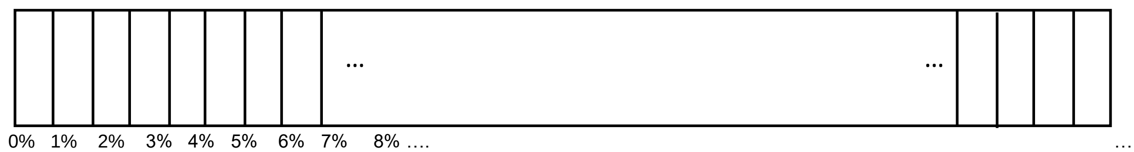

Figure 5. Discount ladder of items

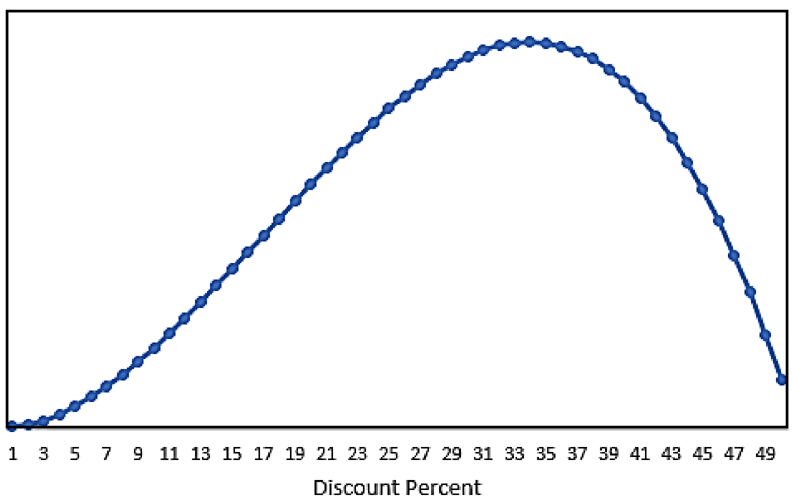

Figure 6. Effect of discount percent on profit

The discount optimizer algorithm starts the search procedure from 0% of discounts for all products and pass a vector of zeros to neural network and calculates the profit. After this initialization the profit of each product is stored in a vector. After this initialization, the next steps are as follows:

1. Discount percent of each product increases by 1%, since the discount ladder allows to increase the discounts and movie on the next step.
2. Calculate the profits.
3. Each product which has increase in profit, is a candidate for moving on next step of ladders and other ones stop on moving through ladder.
4. The steps continue to all products are eliminated.

Results

Dataset

For ease of explanation, assume the dataset under study $[\boldsymbol{\psi}]_{\boldsymbol{R}\times\boldsymbol{N}\times\boldsymbol{F}}$ contains $\boldsymbol{R}$ records (each record belongs to one day), where the number of features for each record equals $\boldsymbol{F}$. Additionally, the total number of products to be modeled and forecasted is considered as $\boldsymbol{N}$. Furthermore, the vector $\boldsymbol{r}_i^n$, $\boldsymbol{i} \in \{\boldsymbol{1}, \boldsymbol{2}, \ldots \boldsymbol{R}\}, \boldsymbol{n} \in \{\boldsymbol{1}, \boldsymbol{2}, \ldots \boldsymbol{N}\}$ represents the feature vector of the $\boldsymbol{n^{th}}$ product on the $\boldsymbol{i^{th}}$ day and actually includes all features affecting the demand for that product. Therefore, $\boldsymbol{r}_i^n$ will be a vector of length $\boldsymbol{F}$ where each element represents the value of one feature. Regarding demand values, the dataset $[\boldsymbol{D}]_{\boldsymbol{R}\times\boldsymbol{N}}$ is considered, where $\boldsymbol{d}_i^n$, $\boldsymbol{i} \in \{\boldsymbol{1}, \boldsymbol{2}, \ldots \boldsymbol{R}\}, \boldsymbol{n} \in \{\boldsymbol{1}, \boldsymbol{2}, \ldots \boldsymbol{N}\}$ represents the demand for the $\boldsymbol{n^{th}}$ product on the $\boldsymbol{i^{th}}$ day. In summary, considering $\boldsymbol{R}$, $\boldsymbol{N}$ and $\boldsymbol{F}$ as the number of records, number of products, and number of features respectively, the data tensor corresponds to the cube shown in Figure 7.

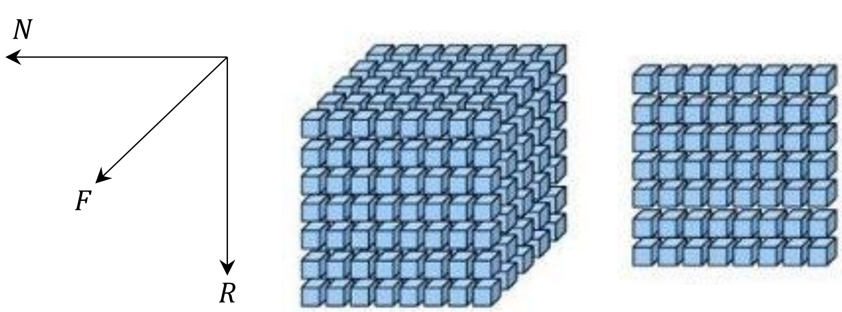

Figure 7. Data tensor

Preprocessing and feature extraction

The dataset under study belongs to one of the country's chain stores, collected over 14 months from 18 provinces. It contains product sales information, and its features will be described in detail later. Data preprocessing involves handling missing values and outliers. In both cases, the correction or imputation of values will be performed using the $\boldsymbol{k}$-nearest neighbors regression algorithm (Martínez et al., 2019). To explain this algorithm, assume the vector $[\boldsymbol{r}_i^n, \boldsymbol{d}_i^n]_{\mathbf{1}\times(\boldsymbol{F}+\mathbf{1})}$ contains the features and demand of the $\boldsymbol{n}^{\boldsymbol{th}}$ product on the $\boldsymbol{i}^{\boldsymbol{th}}$ day, with some of its components missing or being outliers. In this case, based on the other components, its $\boldsymbol{k}$-nearest neighbors are identified, and the missing or outlier values are replaced with the average of these neighbors. If a component is discrete, the rounded average will be used instead.

All computations and modeling will be performed on the dataset collected from a chain store over 14 months. The data is categorized by province, with multiple stores in each province. Some stores may be inactive on certain days due to inventory audits or other factors. The granularity of the data is daily, with sales records for multiple products available each day. The features included in the dataset are listed in Table 2, considering the feature stock level at the table is an array in which the element $\boldsymbol{i}^{\boldsymbol{th}}$ is an integer number refers the stock level of $\boldsymbol{i}^{\boldsymbol{th}}$ store.

Table 2. Features of raw dataset

Feature	Type
Product ID	Numeric
Product Name	String
Province Name	String
Date	Numeric
Number of Stores	Numeric
Number of Active Stores	Numeric
Holiday	Binary
Discount	Percentage
Stock Level	Vector

According to what is available in the dataset, additional features such as demand-influencing occasions and inventory scores can be extracted. The occasions are derived based on the information in Table 4 and will be included in the dataset as dummy variables.

Table 3. Dummy Variables for Calendar Occasions Used in the Dataset

Feature	Description
Day of Month	Consists of 31 binary components (1 to 31), where only the component corresponding to the i^{th} day of the solar month equals 1, and all others are 0.
Week of Month	Includes 5 components (one for each week of the month), where the component corresponding to the current week is set to 1, and the rest are set to 0.
Day of Week	Includes components for Saturday to Friday, where the component corresponding to the current day is set to 1.
Month of Year	Includes 12 components (Farvardin to Esfand), where the component corresponding to the current month is set to 1.
Holiday	A binary variable set to 1 if the current day is a holiday, otherwise 0.
Before Holiday	A binary variable set to 1 if the next day is a holiday, otherwise 0.
After Holiday	A binary variable set to 1 if the previous day was a holiday, otherwise 0.
Successive Holidays	An integer indicating the number of consecutive holidays up to the current day (e.g., a value of 2 means both the previous day and the current day are holidays).
Nowruz (Persian New Year)	Set to 1 during the 13-day Nowruz period, otherwise 0.
13-be-dar	Set to 1 only on the 13th of Farvardin (Sizdah Bedar), otherwise 0.
Muharram (Ashura Period)	Set to 1 during the first 10 days of Muharram, otherwise 0.
Tasua, Ashura and Arba`en	Set to 1 in three days of Tasua, Ashura and Arba`en, otherwise 0.
Ramadan Period	Set to 1 during the month of Ramadan, otherwise 0.
Before Ramadan	Set to 1 on the three days before Ramadan begins, otherwise 0.
Eid al-Fitr	Set to 1 on Eid al-Fitr, otherwise 0.
End of Year	Set to 1 on the last three days of the year, otherwise 0.

Numerical Experiments

At the beginning of this section, the correlation between the inventory level score and product sales is examined. This score is calculated on a scale from 0 to 1, where 0 indicates the product is out of stock in all stores, and 1 indicates that more than 20 units are available in all stores (see Table 1). Values between 0 and 1 can be interpreted accordingly. Figure 6 illustrates the correlation pattern for four sample products, demonstrating that incorporating this feature into the model can improve its performance and prediction accuracy.

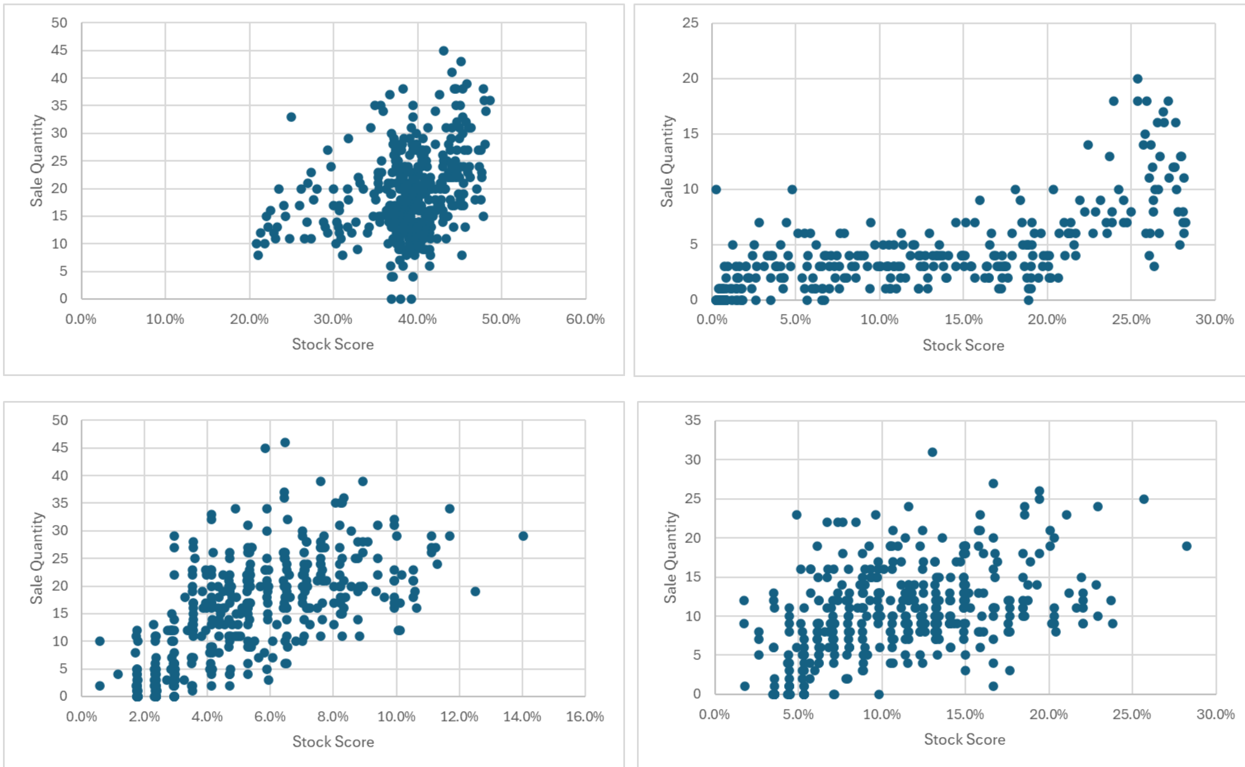

Figure 8. Correlation of stock level score and sales

As outlined in the flowchart depicted in Figure 4, the PSO algorithm was employed to optimize the parameters of the neural network. The initial parameters were configured according to Table 4. Additionally, the activation function selection was delegated to the optimization algorithm as specified in Table 3. In each iteration of the PSO algorithm, the neural network was trained using the Adam algorithm (pseudocode shown in Figure 5). The training-to-test data ratio was set at 80% and 20%, respectively. This ratio ensures that approximately one year's worth of data is included in the neural network's training phase, thereby incorporating all seasonal behaviors, promotions, holidays etc. over a full annual cycle into the model's training. At each stage, the Mean Absolute Error (MAE) of the test data was used to evaluate the model's performance.

Table 4. PSO parameters for neural network training

Parameters		Parameter Category		Value
Number of Particles		PSO		20
Number of Iteration		PSO		50
c_1		PSO		1.4
c_2		PSO		1.4
Maximum Number of Hidden Layers		Neural Network		32
Maximum Number of Neurons in Each Layer		Neural Network		128
Epoch		Neural Network		50
Batch Size		Neural Network		32

For example, Figure 9 demonstrates the convergence of the MAE index over 20 iterations of the PSO algorithm. After the 10th iteration, the value remained unchanged, and the algorithm maintained this value until completion. Through successive iterations, the algorithm successfully reduced the MAE index from 5.33 to 4.95. Figure 8 additionally displays the loss function and neural network's MAE index during one of the initial iterations of the PSO algorithm.

To report the obtained metrics and demonstrate the impact of the inventory level score feature, please refer to Table 5. This table examines seven sample products under two conditions: without parameter tuning and with PSO-optimized parameter tuning. In all cases, parameter optimization improved the model's performance.

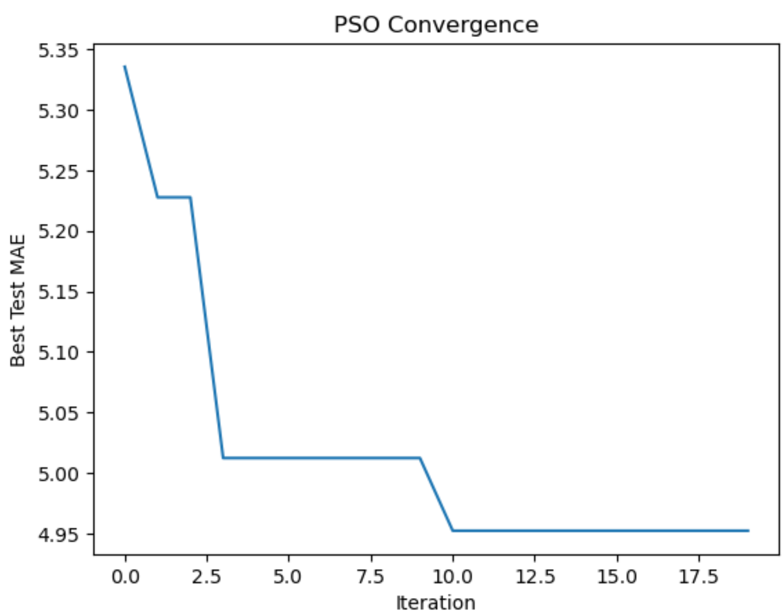

Figure 9. Converging MAE measure of neural network in PSO iterations

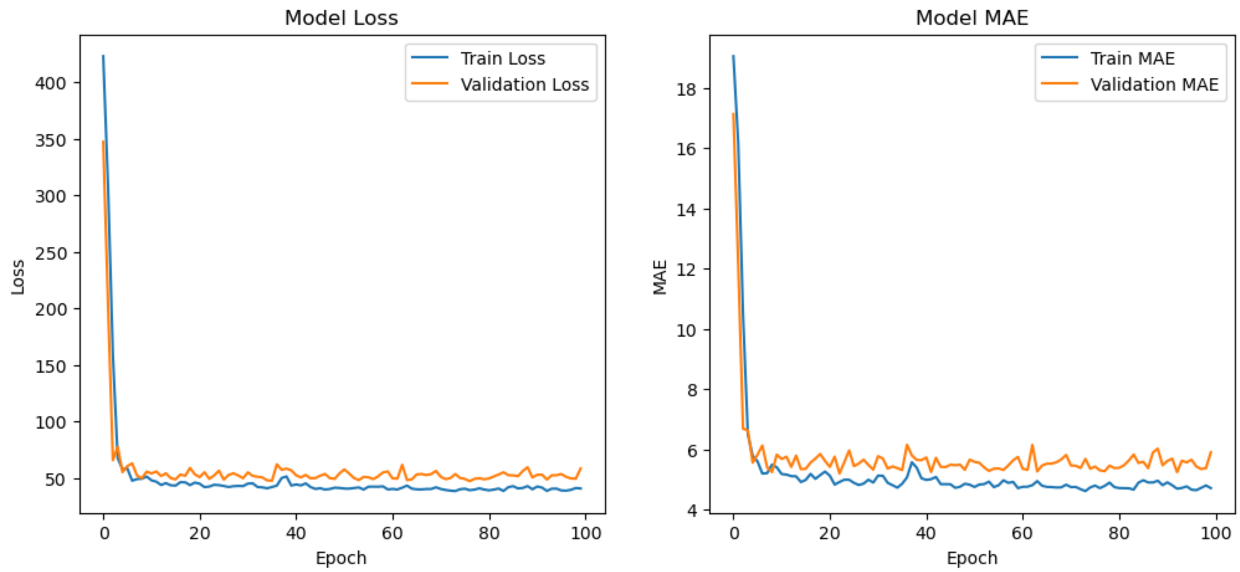

Figure 10. Loss function and MAE of neural network in one of PSO iterations

Table 5. MAE measure for test dataset

Product ID	Random Parameters		Parameter Tunning With PSO	
	With Stock Level feature	Without Stock Level Feature	With Stock Level feature	Without Stock Level Feature
1	5.33	6.73	4.96	6.00
2	5.48	7.23	4.44	5.30
3	0.89	1.23	0.86	1.00
4	9.26	11.03	5.05	7.70
5	8.29	10.00	6.97	8.29
6	6.14	9.29	4.43	8.08
7	12.18	13.14	12.07	7.93
8	8.55	8.69	7.04	12.47
9	13.82	14.67	12.56	12.24
10	10.91	11.64	11.53	17.13
11	15.64	15.51	15.44	2.16
12	1.36	2.21	1.22	13.87
13	15.82	15.62	13.82	10.75
14	10.36	10.79	9.23	12.05
15	13.00	13.40	10.82	10.15
16	9.64	9.94	10.16	12.09
17	15.82	16.90	11.16	15.07
18	15.18	16.61	14.37	1.41
19	1.82	1.79	1.01	3.08
20	7.24	13.47	5.14	8.01

Moreover, excluding the inventory level score feature results in degraded MAE performance, demonstrating that this variable contributes significantly to explaining product sales variations.

To evaluate the effectiveness of the stock-level feature and the parameter tuning performed using the PSO algorithm, a hypothesis test was conducted. First, to assess the contribution of the stock-level feature, the MAE values obtained under random parameter settings were compared (i.e., the first two columns of Table 5). The resulting p-value 0.0078125 indicates a statistically significant reduction in MAE when the stock-level feature is incorporated.

Similarly, a comparison between the MAE values obtained using random parameters with the stock-level feature and those obtained using PSO-tuned parameters with the stock-level feature yielded a p-value of 0.0081. This result confirms that PSO-based parameter tuning further improves model performance.

To optimize discounts, the algorithm of section 5-3 is employed. Starting from the first step of discount ladder, 0%, Figure 9 illustrated the results of products on each step. It is obvious that when the algorithm proceeds to next iteration, some products are given up and do not continue on the ladder. Finally, on 42th step of the ladder which is equivalent to 42% percent of discount all products are dopped, meaning that the maximum discount percent is 42%.

From profit viewpoint, it is obvious that during the iterations of the algorithm profit sill be increased. Figure 11 reveals this behavior.

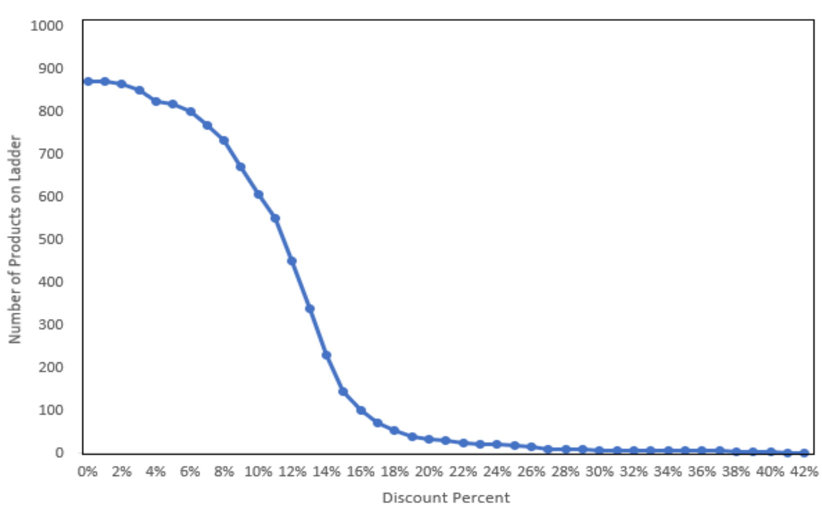

Figure 11. Number of products on later based on different discounts

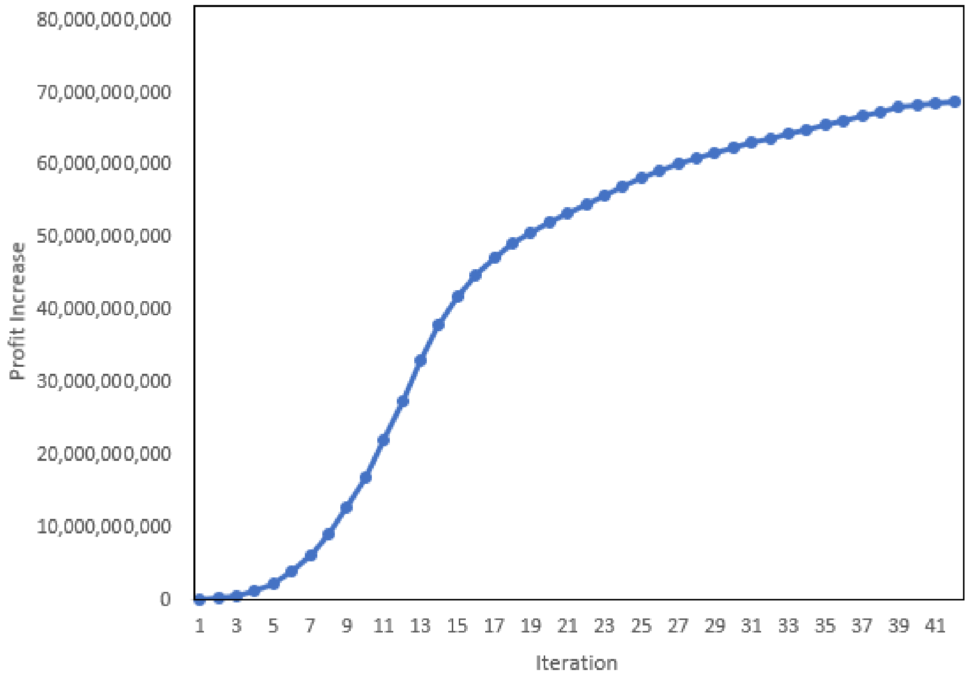

Figure 12. Profit increase per each iteration of algorithm

Conclusion

This paper presented a framework for integrating sales forecast and discount optimization in retail sector. In sales forecasting phase, stock level score is formulated to be used as a feature beside the calendar ones and discount percentage in deep neural network to achieve more precise forecasts. The model parameters including activation functions and number of neurons in each layer were optimized using the PSO algorithm. In addition, a discount optimizer employs the demands to maximize total profit based on discount ladder approach. The main idea is to increase discount percentage of each item by a pre-defined step size on discount ladder until the best combination of discounts is obtained. The computational results focused on forecasting precision by mean absolute of errors as well as the total profit. Numerical experiments revealed that stock level feature leads to MAE improvement in most products. Parameter tunning in deep network topology affects the performance of model. In the case of discount optimization, the outcome of forecasting model is used to maximize the profit. Moving on discount ladder can tune appropriate discount percentage of products, where profit enhancement is obvious according to results. Overall, the proposed framework demonstrates the effectiveness of integrating demand forecasting and pricing decisions within a unified model. The results indicate that the interaction between inventory information and discount strategy plays a significant role in improving retail profitability. These findings highlight the practical applicability of the proposed approach for data-driven decision support in retail operations.

Feature study

While the proposed discount optimizer employs a greedy, SKU-wise ladder search to ensure scalability and operational simplicity, it does not explicitly account for cross-elasticities, substitution effects, or basket-level interactions across products. In real retail environments, price changes in one SKU may influence the demand of related items, affecting overall category performance. Future research could extend the framework toward a multi-product optimization setting by incorporating cross-price elasticity estimation, transaction-level basket modeling, or graph-based product relationship structures. Integrating such interdependencies into a joint optimization model (potentially through mixed-integer programming or reinforcement learning approaches) would enable coordinated discount decisions at the category level and may yield further improvements in overall profitability and demand forecasting accuracy.

Data Availability Statement

Data available on request from the authors.

Acknowledgements

The authors gratefully acknowledge the retail company which provided data for this research.

Conflict of interest

The authors declare no conflict of interest related to the content, data, or results of this study. All analyses were conducted independently, and no external parties influenced the interpretation or presentation of the findings.

Funding

The authors received no financial support for this article's research, authorship, and/or publication.

Ethical considerations

The authors declare compliance with all applicable ethical guidelines, including proper data handling, originality of content, and avoidance of duplicate submission.

References

Aktas, M. Y., Ji, T., & Lu, C. T. (2024, December). Time series forecasting with GCN-LSTM based unified model for product demand prediction. In 2024 IEEE International Conference on Big Data (BigData) (pp. 5901–5906). IEEE. https://doi.org/10.1109/bigdata62323.2024.10825969

Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*, 112896. https://doi.org/10.1016/j.eswa.2019.112896

Bergsma, R., de Ruijt, C., & Bhulai, S. (2025). A systematic review of machine learning approaches in inventory control optimization. *Operations Research Perspectives*. https://doi.org/10.1016/j.orp.2025.100367

Bharadwaj, A. A., & Gunasekaran, M. (2025). Forecasting of product demand using hybrid regression model in comparison with autoregressive integrated moving average model. In Information Visualization for Intelligent Systems (pp. 17–28). https://doi.org/10.1002/9781394305810.ch2

Cabbalero, S., & Rice, J. (2018). Artificial intelligence and machine learning in supply chain planning. *MIT Center for Transportation & Logistics*. https://doi.org/10.4018/979-8-3693-4433-0.ch014

Crockett, K., Bandar, Z., & O'Shea, J. (2009, August). Fuzzification of discrete attributes from financial data in fuzzy classification trees. In 2009 IEEE International Conference on Fuzzy Systems (pp. 1320–1325). IEEE. https://doi.org/10.1109/fuzzy.2009.5277400

Dave, E., Leonardo, A., Jeanice, M., & Hanafiah, N. (2021). Forecasting Indonesia exports using a hybrid model ARIMA-LSTM. *Procedia Computer Science*, *179*, 480–487. https://doi.org/10.1016/j.procs.2021.01.035

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Fisher, M., Rajaram, K., & Raman, A. (2001). Optimizing inventory replenishment of retail fashion products. *Manufacturing & Service Operations Management*, *3*(3), 230–241. https://doi.org/10.1287/msom.3.3.230.9836

Gaur, V., Fisher, M. L., & Raman, A. (2005). An econometric analysis of inventory turnover performance in retail services. *Management Science*, *51*(2), 181–194. https://doi.org/10.1287/mnsc.1040.0298

Gumte, K. M., Pantula, P. D., Miriyala, S. S., & Mitra, K. (2021). Data driven robust optimization for handling uncertainty in supply chain planning models. *Chemical Engineering Science*, *246*, 116889. https://doi.org/10.1016/j.ces.2021.116889

Horden, H. E., & Bisset, C. (2022). Optimisation of promotions in the retail industry. *South African Journal of Industrial Engineering*, *33*(3), 40–54. https://doi.org/10.7166/33-3-2470

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, *36*(4), 1420–1438. https://doi.org/10.1016/j.ijforecast.2020.02.005

Ji, L., Zou, Y., He, K., & Zhu, B. (2019). Carbon futures price forecasting based with ARIMA-CNN-LSTM model. *Procedia Computer Science*, *162*, 33–38. https://doi.org/10.1016/j.procs.2019.11.254

Joseph, R. V., Mohanty, A., Tyagi, S., Mishra, S., Satapathy, S. K., & Mohanty, S. N. (2022). A hybrid deep learning framework with CNN and bi-directional LSTM for store item demand forecasting. *Computers & Electrical Engineering*, *103*, 108358. https://doi.org/10.1016/j.compeleceng.2022.108358

Kazemian, M., Afsharkazemi, M. A., Fathi Hafashjani, K., & Motadel, M. (2023). Presenting smart steel pricing model: An integration of game theory and machine learning algorithms. *Industrial Management Journal*, *15*(3), 478–507. https://doi.org/10.22059/imj.2023.356697.1008039

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. https://arxiv.org/abs/1412.6980

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv. https://arxiv.org/abs/1707.01926

Liu, J., Wang, G., Zhao, H., Lu, M., Huang, L., & Chen, G. (2025). Beyond complements and substitutes: A graph neural network approach for collaborative retail sales forecasting. *Information Systems Research*.

Martínez, F., Frías, M. P., Pérez, M. D., & Rivera, A. J. (2019). A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, *52*(3), 2019–2037. https://doi.org/10.1007/s10462-018-9659-3

Meisheri, H., Sultana, N. N., Baranwal, M., Baniwal, V., Nath, S., Verma, S., Ravindran, B., & Khadilkar, H. (2022). Scalable multi-product inventory control with lead time constraints using reinforcement learning. *Neural Computing and Applications*, *34*(3), 1735–1757. https://doi.org/10.1007/s00521-021-06575-4

Mughees, N., Mohsin, S. A., Mughees, A., & Mughees, A. (2021). Deep sequence to sequence bi-LSTM neural networks for day-ahead peak load forecasting. *Expert Systems with Applications*, *175*, 114844. https://doi.org/10.1016/j.eswa.2021.114844

Nagai, S., Inaba, R., Oishi, R., Aikawa, S., Mibuchi, Y., Moriyama, H., … & Nakata, K. (2024, December). Zero-shot demand forecasting for products with limited sales periods. In 2024 IEEE International Conference on Big Data (BigData) (pp. 5154–5160). IEEE. https://doi.org/10.1287/isre.2023.0773

Neslin, S. A., Grewal, D., Leghorn, R., Shankar, V., Teerling, M. L., Thomas, J. S., & Verhoef, P. C. (2006). Challenges and opportunities in multichannel customer management. *Journal of Service Research*, *9*(2), 95–112. https://doi.org/10.1177/1094670506293559

Reid, R. D., & Sanders, N. R. (2019). Operations management: An integrated approach. John Wiley & Sons. https://doi.org/10.1108/01443570610650585

Rezasoltani, A., Jafarnejad Chaghoshi, A., & Khani, A. M. (2024). A voting-based hybrid machine learning model for predicting backorders in the supply chain. *Industrial Management Journal*, *16*(3), 457–481. https://doi.org/10.22059/imj.2024.381521.1008178

RiachY, C., He, M., Joneidy, S., et al. (2025). Enhancing deep learning for demand forecasting to address large data gaps. *Expert Systems with Applications*, *268*, 126200. https://doi.org/10.1016/j.eswa.2024.126200

Seyam, A., Barachi, M. E., Mathew, S. S., Du, B., & Shen, J. (2024, November). Machine learning-driven daily demand forecasting for fresh produce: A case study with bananas. In 2024 Twelfth International

Conference on Advanced Cloud and Big Data (CBD) (pp. 363–368). IEEE. https://doi.org/10.1109/cbd65573.2024.00071

Sütçü, M., & Yıldız, B. (2025). A stochastic programming framework for pricing and market share optimization in retail systems. *Decision Analytics Journal*, *16*, 100604. https://doi.org/10.1016/j.dajour.2025.100604

TaghizadehYazdi, M. R., Mirshojaeian Hosseini, H., Asgharizadeh, E., & Shakouri Ganjavi, H. (2015). Relative performance of statistical and fuzzy regression models in estimation of gasoline demand in Iran. *Industrial Management Journal*, *7*(1), https://doi.org/10.22059/imj.2015.50684

Tang, C. S., Rajaram, K., Alptekinoğlu, A., & Ou, J. (2004). The benefits of advance booking discount programs: Model and analysis. *Management Science*, *50*(4), 465–478. https://doi.org/10.1287/mnsc.1030.0188

Trapero, J. R., de Frutos, E. H., & Pedregal, D. J. (2024). Demand forecasting under lost sales stock policies. *International Journal of Forecasting*, *40*(3), 1055–1068. https://doi.org/10.1016/j.ijforecast.2023.10.002

Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. arXiv. https://arxiv.org/abs/1711.11053

Wu, M., Demirag, O. C., Xue, W., & Xu, M. (2024). Retail category management under shelf-space dependent demand: The effectiveness of category captainship. *International Journal of Production Economics*, *276*, 109365. https://doi.org/10.1016/j.ijpe.2024.109365

Yang, H., Huang, T., & Chen, K. (2025). Marrying theory and practice: Product network-inspired deep learning for sales forecasting. SSRN. https://ssrn.com/abstract=5118746