EURO Journal on Transportation and Logistics 12 (2023) 100103

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

Optimizing inventory control through a data-driven and model-independent framework

Evangelos Theodorou *, Evangelos Spiliotis, Vassilios Assimakopoulos

Forecasting and Strategy Unit, School of Electrical and Computer Engineering, National Technical University of Athens, Greece

ARTICLE INFO

Keywords:
Inventory control management
Machine learning
Inventory cost
Demand patterns
Large scale optimization

ABSTRACT

Machine learning has shown great potential in various domains, but its appearance in inventory control optimization settings remains rather limited. We propose a novel inventory cost minimization framework that exploits advanced decision-tree based models to approximate inventory performance at an item level, considering demand patterns and key replenishment policy parameters as input. The suggested approach enables data-driven approximations that are faster to perform compared to standard inventory simulations, while being flexible in terms of the methods used for forecasting demand or estimating inventory level, lost sales, and number of orders, among others. Moreover, such approximations can be based on knowledge extracted from different sets of items than the ones being optimized, thus providing more accurate proposals in cases where historical data are scarce or highly affected by stock-outs. The framework was evaluated using part of the M5 competition's data. Our results suggest that the proposed framework, and especially its transfer learning variant, can result in significant improvements, both in terms of total inventory cost and realized service level.

1. Introduction

Most retailers, even medium and small-sized ones, need to manage thousands of unique items at a regular basis to minimize operational costs and maximize sales (Seaman, 2018). A critical part of the managing process refers to inventory control, i.e. defining whether and when an order should be placed for a particular item, as well as how many units such an order should involve. Although shorter review periods and larger orders typically increase product availability and reduce the probability of losing sales, they also have a negative impact on inventory costs, including stock holding and ordering costs, among others. To deal with such a challenging trade-off, inventory policies have to be carefully optimized in terms of parameters (e.g. review period, lead time, and target service level) for each item separately.

While numerical approaches, such as dynamic programming based methods (Lagodimos et al., 2012; Tsai and Chen, 2017), can be used to support such decisions, when the dimensions of the optimization problem grow (e.g. in terms of number of series and sets of inventory policy parameters), these techniques can become extremely time consuming, rendering them obsolete. Moreover, most of these approaches are not data-driven in nature, making strong assumptions about the distribution of the demand and the replenishment process used, being also negatively affected by possible data availability issues and effects that stock-outs or delivery delays may imply.

A promising alternative to the numerical approaches are machine learning (ML) algorithms. These algorithms have the ability to generalize non-linear processes by learning from large data sets and applying the knowledge acquired for making predictions or recommendations. Equally important, ML algorithms make no or little assumptions about the data, thus establishing their recommendations on the patterns observed instead of prescribing the data generation process. The utilization of ML algorithms has shown excellent performance in various areas of the supply chain management. For instance, ML algorithms have been proposed as an alternative to statistical methods for forecasting demand data (Spiliotis et al., 2020; Makridakis et al., 2022), considered for classifying inventory items (Lolli et al., 2019), and used for predicting optimal routes in transportation problems (Zimmermann and Frejinger, 2020), among others.

In the field of stock control optimization, Sustrova (2016) used various types of neural networks to optimize the order sizes of a company's products under the periodic review inventory policy, considering as inputs the demand, the forecasted demand of the following months, the current inventory level, as well as the purchase and transportation costs of each product. Inprasit and Tanachutiwat (2018) compared different training algorithms for constructing a neural network that calculates the optimal safety stock and reorder point of a product using as inputs the lead time and its variance, the demand and its variance,

* Corresponding author.
E-mail addresses: vagtheodorou@fsu.gr (E. Theodorou), spiliotis@fsu.gr (E. Spiliotis), vassim@fsu.gr (V. Assimakopoulos).

https://doi.org/10.1016/j.ejtl.2022.100103
Received 1 March 2022; Received in revised form 6 September 2022; Accepted 14 December 2022
Available online 27 December 2022
2192-4376/

Nomenclature

Variables

i	Product index (i=1, …, N)
t	Period index (t=1, …, T)
$\hat{D}_{i,t}$	Forecasted demand of product i for period t
$D_{i,t}$	Effective demand of product i for period t
$s_{i,t}$	Order point of product i for period t
$S_{i,t}$	Order-up-to level of product i for period t
$SS_{i,t}$	Safety stock of product i for period t
$I_{i,t}$	Inventory position of product i for period t
$Q_{i,t}$	Number of items i ordered in period t
m	Periods covered by a replenishment
C_H	Holding inventory cost over T periods
C_L	Lost sales costs over T periods
C_O	Ordering cost over T periods
C_{Tot}	Total inventory cost over T periods
$I_{i,t}$	Inventory level of product i for period t
$\overline{I}_i$	Average inventory level of product i over T periods
$LS_{i,t}$	Lost sales of product i for period t
$NO_{i,t}$	Number of orders placed for product i on period t (1 or 0)
SL_i	Realized service level of product i
$\overline{SL}$	Average realized service level
Z	z-score calculated using the inverse cumulative distribution function of the normal distribution

Inventory policy hyperparameters

R_i	Review period (in days) of product i
L_i	Lead time of product i
TSL_i	Target service level of product i

Inventory cost hyperparameters

p_i	Unit value of product i
h	Annual unit holding charge, expressed as a percentage of the product value
k	Cost to place an order
b	Unit shortage cost, expressed as a percentage of the product value

Time series features

ADI_i	Average inter-demand interval of product i
CV_i^2	Squared coefficient of variation of demand of product i
$\overline{D}_i$	Average demand of product i over T periods

and the target customer service level, concluding that the Bayesian regularization backpropagation has the lowest prediction error. Namir et al. (2022) proposed a framework that combines statistical time series forecasting methods, machine learning algorithms, and combinatorial optimization to identify opportunities about buying inventory at a lower cost and selling or holding a portion of that inventory with the objective to maximize profits. Yet, despite these encouraging applications, according to Boute et al. (2021) the appearance of ML algorithms in inventory control settings has remained rather limited, with further research in the field being strongly recommended (Simchi-Levi, 2014; Mišić and Perakis, 2020).

In this respect, we propose a data-driven framework that can be used to optimize replenishment at warehouse level. With our approach, a reference set of items is used to learn how inventory performance may be affected by the patterns of the demand and key inventory policy parameters. Then, this knowledge is exploited to make accurate recommendations about the inventory policy parameters that should be used to optimize the replenishment of similar, target series. To do so, the costs implied by lost sales, holding stock, and ordering are taken into consideration. Effectively, the reference series may not be the same as the target ones, thus enabling transfer learning and optimizations in cases where historical observations are limited. The proposed framework is model-independent, building on machine learning and forecasting methods of preference.

We evaluate the proposed approach using part of the M5 competition data (Makridakis et al., 2022) and consider indicative benchmarks to investigate possible improvements. Our approach results in significant improvements, both in terms of total inventory cost and realized service level.

The contributions and innovations of our paper are summarized as follows:

- We introduce a data-driven approach as an alternative to numerical approaches to address the problem of inventory policy optimization. Our approach is generic and flexible in nature, being applicable to complex policies and making little assumptions about the demand distributions. As a result, it can be easily modified in terms of parameters to best fit the application at hand.
- We exploit advanced ML methods, namely gradient boosted regression tree models, to approximate various aspects of the inventory performance (e.g. cost and service level), taking into consideration the characteristics of the demand and key replenishment policy parameters.
- We empirically evaluate the performance of the proposed approach by conducting excessive experiments on a large, real world data set of the retail industry and using indicative benchmarks.
- We showcase that, once the training phase of the ML models is finished, our approach can approximate inventory performance in a fraction of the time than a standard, analytical method would require. This finding renders our approach a powerful tool for fast optimization and decision making.
- We illustrate that the proposed ML models can be applied in a transfer learning fashion with excellent results, a finding that enables their exploitation at a large scale, even by small-size retail companies that may have neither the data nor the experience required for training and implementing the models in a standard fashion.

The rest of the paper is organized in five sections. Section 2 describes the background of the problem, also connecting inventory control policies with demand categorization schemes and forecasting methods. Section 3 introduces the proposed framework at a methodological basis and explains how it will be implemented in particular in the present study. Section 4 presents the experimental setup, including the data and benchmarks used for the empirical evaluation. Finally, Section 5 summarizes and discusses our results, while Section 6 concludes the paper.

2. Inventory management

Significant research has been conducted to improve inventory management in businesses, covering applications ranging from distribution planning to demand forecasting and inventory level control (Syntetos et al., 2010). Due to the complexity of these applications and their influence by multiple factors, both internal and external to the company, each inventory management process is typically optimized separately, considering the particular characteristics of the firm and its environment, the strategic objectives set, and the measures used for evaluating performance, among others.

When it comes to replenishment, optimization can be rarely performed in a prescriptive fashion, rendering the whole process into a challenging problem. Depending on the pattern of the demand, the codependency of the items, the stock control policy used, the restrictions applied to order quantities and deliveries, the inventory performance measures adopted, and the costs that ordering, holding inventory, and losing sales imply, different decisions would be considered as "optimal". The following subsections summarize the key variables that have to be defined to improve replenishment, as well as the assumptions typically made to facilitate the whole process.

2.1. Categories of demand

Understanding the patterns of the demand is one of the most critical factors for optimizing replenishment. The demand of each item may depict different characteristics in terms of variance and intermittency, among others, meaning that different methods may be more appropriate for forecasting each item. Accordingly, different strategies may be more effective for undertaking replenishment for each case (Boylan et al., 2008).

Syntetos et al. (2005), expanding on previous work done in the field (Williams, 1984; Johnston and Boylan, 1996; Eaves and Kingsman, 2004), categorized demand in four classes based on the average inter-demand interval (ADI) and the squared coefficient of variation of the demand sizes (CV^2) of the time series, associating each category with particular forecasting methods.

ADI summarizes the intermittency of demand, i.e. whether the demand is sporadic including periods of zero demand, and is calculated as follows:

$$ADI = \frac{\sum_{i=1}^{N} t_i}{N}, \tag{1}$$

where N is the number of periods with non-zero demand and t_i is the interval between two consecutive non-zero demands. An ADI value of 1 suggests lack of intermittency, while larger values the existence of zero demands. CV^2 summarizes the variability of the demand when it occurs and is calculated as follows:

$$CV^2 = \left(\frac{\sigma_D}{\overline{D}}\right)^2 = \frac{\sum_{i=1}^{N}(D_i - \overline{D})^2}{(N-1)\overline{D}^2}, \tag{2}$$

where D_i and $\overline{D}$ are the i_{th} non-zero demand and the average non-zero demand, respectively. Larger CV^2 values suggest higher volatility of demand.

Based on the criteria above, according to Syntetos et al. (2005) demand time series can be categorized as "smooth", "erratic", "lumpy", and "intermittent", with the corresponding thresholds (cut-off values) for separating the four categories being 0.49 for CV^2 and 1.32 for ADI, as shown in Fig. 1. This categorization scheme, expanding on previous work by Williams (1984), Eaves and Kingsman (2004), and Johnston and Boylan (1996), correlates key demand patterns with the accuracy of three popular demand forecasting methods, namely the Croston's method (Croston, 1972), the Syntetos–Boylan approximation (SBA; Syntetos and Boylan, 2005), and the simple exponentially weighted moving averages. Although in principle considering a set of different or additional forecasting methods may lead to different cut-off values, Syntetos et al. (2005) support the stability of the proposed thresholds that have been widely adopted in the demand forecasting literature, including the analysis of the M5 competition results. To that end, these thresholds were also used in the present study to facilitate the discussion of our results.

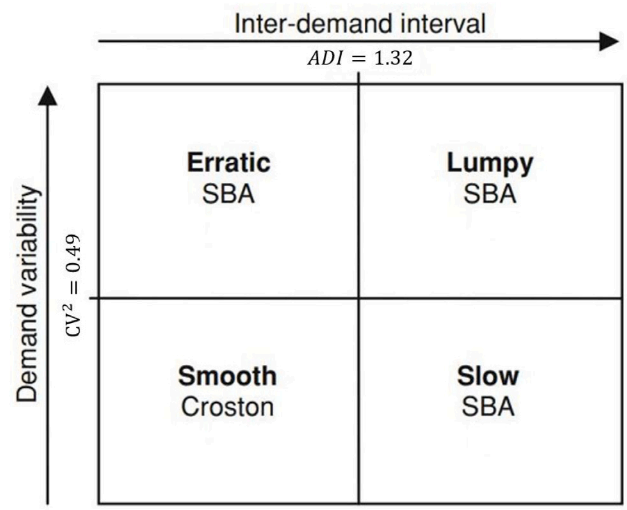

Fig. 1. Categories of demand patterns and best performing forecasting methods per case, as proposed by Syntetos et al. (2005).

2.2. Stock control policies

Assuming the demand of an item is constant and deterministic, the economic order quantity (EOQ) method (Harris, 1913) can be used to estimate the order quantity Q that reduces the costs associated with inventory carrying and ordering at period t, as follows:

$$Q = \sqrt{\frac{2D_{year}k}{H}}, \tag{3}$$

where D_{year} is the annual demand quantity, k is the fixed cost per order, and H is the annual holding cost per unit.

However, in practice, the aforementioned assumptions about the demand are rarely satisfied, rendering the results of the EOQ method misleading. In most cases, where demand is not only stochastic but also non-stationary (the distribution of the demand changes form as time passes), replenishment quantity can be estimated more accurately by forecasting the demand over a planning horizon using the available historical data. Since forecasting accuracy and horizon can substantially affect costs, forecasting and inventory policies are strongly connected.

Various policies can be used to control stock of items at a single location. Common classifications of these policies are based on whether the review period is continuous or periodic, as well as whether the order quantity is constant or varying. The continuous (s, Q) policy is probably the most popular one, suggesting that when the inventory level gets below the order point s, an order Q should be placed. The periodic (R, S) policy is also common, suggesting that every R periods an order-up-to level S should be placed. Another stock control policy, that has been proposed for intermittent demand settings and is often encountered in practice (Silver et al., 1998; Syntetos and Boylan, 2006), is the (R, s, S) policy, in which at review moments, R time-units apart, and if the inventory level at that moment is at or below the reorder point s, an order-up-to level S is placed (see Goltsos et al., 2022 for an in-depth literature review).

When the stock level is insufficient for fulfilling an order, the policy may either assume that the order will be satisfied as soon as some stock becomes available (backorders) or that the corresponding sales will be lost. In the first scenario, where customers eventually receive their orders and no sales are lost, a cost associated with said delay is typically taken into account (waiting time may affect the goodwill of some customers and decrease the market share of the company in the long term; Anderson et al., 2018). In such settings, order-up-to level policies with periodic reviews are preferred since they are more effective in handling backorders (Scarf, 1960; Federgruen and Zipkin, 1984; Federgruen and Zheng, 1992). However, the backorder assumption may not be applicable in some retail environments, rendering the assumption of lost sales more realistic.

Besides the fact that (R, s, S) policies are proven to be optimal for inventory systems with a backorder assumption (Karlin and Scarf, 1958; Scarf, 1960), they have been hardly studied in a context with lost sales. Moreover, Hill and Johansen (2006) showed that even though this policy is not optimal, in a lost sales context, the expected total inventory costs are close to optimal. Similarly, Bijvank and Vis (2012) calculated that the (R, s, S) policy results in an average cost increase of only 1.1% when fixed ordering costs are considered. The assumptions that any unfulfilled demand is lost and that fixed ordering costs exist are very close to reality, justifying the popularity of the (R, s, S) policy in many practical applications and explaining why we have selected it for the present paper.

Assuming that any unfulfilled demand is lost and that the lead time is constant (the suppliers are perfectly reliable), meaning that the orders are delivered with a fixed delay of length L, the order point and the order-up-to level of the (R, s, S) policy can be calculated using the following set of equations (Strijbosch and Moors, 2002):

$$S_t = max(s_t, \sum_{w=t}^{t+m} \hat{D}_w), \tag{4}$$

$$s_t = \sum_{w=t}^{t+R+L} \hat{D}_w + SS_t, \tag{5}$$

where S_t is the target inventory level at period t, reached through an order placement Q_t, m is the number of periods covered by the replenishment, and s_t the respective order point. At period t, the order point is computed given the expected demand $\hat{D}_w$ for the upcoming periods w, i.e. from t to $t+R+L$, and the safety stock SS. The latter can be calculated using the Greasley's formula (Greasley, 2013) for periodic policy that takes into account the selected probability of no shortage (target service level; TSL) and the standard deviation of historical demand up to period t (σ_{D_t}), as follows:

$$SS_t = Z\sigma_{D_t}\sqrt{R+L}, \tag{6}$$

where Z is the z-score calculated using the inverse cumulative distribution function of the normal distribution, as $Z = \Phi^{-1}(TSL)$.

The order quantity Q_t, which allows the level of inventory (I_t) to reach the target inventory level S_t, is equal to $S_t - I_t$ when the inventory level is lower than the order point s_t and zero elsewhere, as follows:

$$Q_t = \begin{cases} S_t - I_t, & \text{if} \quad I_t \leqslant s_t \\ 0, & \text{if} \quad I_t > s_t. \end{cases} \tag{7}$$

Note that the order point s may be higher than the sum of forecasted demand when $m \geq R + L$ periods. In such cases, the order-up-to level will be equal to the order point.

2.3. Inventory performance measures

Inventory performance can be realized through various measures (Goltsos et al., 2022), typically aiming to the minimization of total costs (cost-based optimization), the convergence to a predefined service level (service-based optimization), or, in most cases, a combination of these two. Measures based on fill rate representations (Heath and Jackson, 1994) and profit functions (Johnston et al., 2011) are other alternatives, although less popular in practice. Note that the fill rate depicts the fraction of demand that is satisfied from the available inventory over multiple consecutive periods, while service level is the probability of no stock-out, calculated as the average fraction of demand that is satisfied at each period separately.

In cost-based optimized policies, the total inventory cost function can be expressed in various ways, including all or part of the costs involved in the replenishment process, starting from the unit cost of the ordered items, the cost of carrying items in inventory, and the cost associated with placing an order, to the costs generated from stock-outs and other costs related with facilitating an order. While in real life these costs may change over time, also frequently depending on the inventory level per se, more often than not, their values are assumed to be constant (Yang, 2014).

By accounting only for the costs emerging from carrying inventory over a period of time, the costs associated with the sales than are not fulfilled, and the costs deriving from placing an order, the total inventory cost (C_{Tot}) over a time horizon of T periods is equal to the sum of holding inventory cost (C_H), lost sales cost (C_{LS}), and ordering cost (C_O), as follows:

$$C_{Tot} = C_H + C_L + C_O = \frac{h}{T}\sum_{i=1}^{N}\sum_{t=1}^{T} I_{i,t}p_i + b\sum_{i=1}^{N}\sum_{t=1}^{T} LS_{i,t}p_i + k\sum_{i=1}^{N}\sum_{t=1}^{T} NO_{i,t}, \tag{8}$$

where h is the annual holding cost per unit, expressed as a percentage of the item value p_i of item i, b is the unit shortage cost, again expressed as a percentage of p_i, and k is the cost to place an order. $I_{i,t}$, $LS_{i,t}$ and $NO_{i,t}$ are the inventory level, lost sales, and number of orders placed at period t for item i, respectively, to be called "inventory cost elements" in the remaining of this paper for reasons of simplicity. In a similar fashion, h, b, and k will be called "inventory cost hyperparameters".

Lambert and Stock (1993) suggest that, depending on the industry, h ranges between 12% and 34%, while more recent studies (Berling and Rosling, 2005; Romer, 1996) support that, for some goods, the dominant part of the holding cost is the capital cost that, on average, can be close to the item value times the real risk-free interest rate, estimated to be around 1%.

Shortage cost consists both of a goodwill loss and lost sales. While lots of research has been conducted on goodwill loss estimation (Alfares and Elmorra, 2005), it is very challenging to perform such approximations in practice, thus being largely omitted (Liao et al., 2011). As a result, b is typically computed by considering only the margin of the items.

Regarding ordering cost, although significant research has been conducted on methods that approximate k based on the lead time (Hemalatha and Annadurai, 2020), it is typically assumed to be fixed (Shang and Zhou, 2010; Altay et al., 2012; Lagodimos et al., 2012), ranging from \$0.25 to \$1.

3. Proposed framework

This section presents a framework that can be used for improving inventory control in companies that have to manage thousands of unique products, each having its own particular characteristics in terms of demand patterns and inventory control elements. Such potential improvements can be realized by better defining the key hyperparameters of the stock control policy employed, namely the review period (R), lead time (L), and target service level (TSL) of an (R, s, S) inventory policy. Instead of utilizing numerical approaches, which are computationally expensive at large scale and infeasible to apply in practice on a regular basis, the proposed framework exploits ML methods that empirically approximate the optimal values of the aforementioned hyperparameters and, as a result, improve performance, both in total and for each item separately. The following subsections provide details on the methods used for forecasting demand, the policy employed for inventory control, the cost function considered for optimizing performance, and the ML methods utilized for making decisions.

3.1. The algorithm

The proposed approach, to be called Model-independent Optimization for Inventory Control ('MOIC'), is presented below and summarized in the flowchart of Fig. 2.

Algorithm: Proposed framework (MOIC).

Off-line phase

Assume K_1 time series (reference set) portraying the demand of K_1 products, each of n historical observations. In the present study, daily time series are used.

1: For each series, time series features are calculated in order to categorize them in terms of demand pattern. In this study, this is done using the ADI and CV^2, as described in Section 2.1.
2: A stock control policy is selected and M sets of inventory policy hyperparameters are generated. In this study, the (R, s, S) inventory policy is selected and 9,000 sets of inventory policy hyperparameters (30 R values × 30 L values × 10 TSL values) are created.
3: A forecasting model of choice, here the Croston's method, is fitted to each series and forecasts are produced.
4: A total inventory cost function is defined, in order to measure the inventory performance for each of the M sets of inventory policy hyperparameters generated at step (2). In this study, Eq. (8) is used as a cost function.
5: Stock control simulations are conducted for a period of a complete calendar year, calculating for each time series and set of inventory policy hyperparameters the respective inventory cost elements (I, LS and NO)
6: A separate ML model, here LightGBM, is built for predicting the inventory cost elements, thus, in our case, resulting in a set of three ML models. The training set of each model consists of $K_1 \times M$ observations and a particular set of features, the number of which depends on the choices made at steps (1) and (2), i.e. time series features and inventory policy hyperparameters. In our case, five features are considered for predicting the target variables.

On-line phase

Assume K_2 time series portraying the demand of K_2 products for which we opt to optimize inventory performance (target series).

7: Time series features, as in step (1), are calculated for the target series.
8: A set of M inventory policy hyperparameter values are selected in a similar fashion to step (2).
9: Using the ML models trained in step (6) and considering the feature values defined in steps (7) and (8) as input, the inventory cost elements are predicted for each series and set of inventory policy hyperparameters.
10: Using the output of step (9) and assuming particular values in terms of inventory cost hyperparameters, in our case h, b, and k, the total inventory cost of the selected setup is estimated per case.
11: Based on the results of step (10), the "optimal" set of inventory policy hyperparameters are identified for each series, according to the cost function selected in step (4).

Note that many elements of the proposed framework can be modified according to the objectives of the decision-maker and the particularities of the retailer. For instance, one may consider different or more time series features for categorizing demand and grouping the items (Theodorou et al., 2022). Similarly, different forecasting methods than the Croston's one can be used for predicting demand and estimating uncertainty. In addition, depending on the inventory policy used by the retailer for replenishment, different inventory policy hyperparameters can be considered for optimization or, in a more generic scenario, various inventory policies can be compared to each other in terms of performance so that the most efficient one is identified. Total inventory cost and the respective hyperparameters can also be adjusted depending on the focus of the examined application. Finally, different ML regression algorithms can be used to model the relationships between the target variables and the features considered as input.

Observe that the series used in the off-line phase of the framework can be either the same or different than the series considered in the on-line phase. In the former case, MOIC will be utilized to optimize replenishment by learning from cost-benefit trade-offs identified in the past for the examined set of items. In the latter case, the replenishment of the target items will be optimized by learning best practices from another set of reference items. This "transfer learning" strategy has recently been proven very effective in the literature for improving forecasting performance (Gautam, 2021; Ye and Dai, 2022), including demand forecasting applications among others (Wellens et al., 2022), especially when the patterns of the reference series (off-line phase) are similar to those of the target series (on-line phase; Ye and Dai, 2021). Apart from improving forecasting performance, transfer learning can also significantly reduce computational cost and data requirements; the models used for conducting the optimization are pre-trained and ready to use for any data set of interest at minor computational cost, enabling optimization even in cases where historical data are scarce for the target series.

3.2. Demand forecasting method

We employ the Croston's method (Croston, 1972) since it is considered the standard method of choice for forecasting intermittent demand time series. The method separates demand into two components, namely the non-zero demand size, D, and the inter-demand intervals, T, and extrapolates them individually using simple exponential smoothing (Gardner, 1985). Both components are forecasted using a smoothing parameter of 0.1 and an initial state value equal to the first observation of each component, as follows:

$$F_i = \frac{\hat{D}_i}{\hat{T}_i}, \text{ where}$$
$$\begin{cases} \hat{D}_i = \hat{D}_{i-1} + a(D_i - \hat{D}_{i-1}) \text{ and} \\ \hat{T}_i = \hat{T}_{i-1} + a(T_i - \hat{T}_{i-1}), \end{cases}$$

where $\hat{D}_i$ and $\hat{T}_i$ are the forecasts of the two components at the i_{th} non-zero demand period, respectively, and F is the final forecast. Note that the forecasts of the Croston's method remain unchanged as far as no demand occurs.

3.3. Inventory policy

The (R, s, S) inventory policy with lost sales is selected for the stock control simulations, utilizing the formulas described in Section 2.2, with a starting inventory level of S. As a result, the inventory policy hyperparameters are the R, L, and TSL.

Although in many cases lead time is considered an exogenous parameter, defined by the supplier and affected by various logistics-related factors, in principle it can be determined by the company, e.g. via a service level agreement (SLA). In this context, and without loss of generality, we assume L to be an internal policy hyperparameter and conduct our experiments accordingly. Similarly, although the selected set of hyperparameters is determined by the (R, s, S) policy considered in our study, in the general case different policies can be assumed, resulting in a different set of hyperparameters.

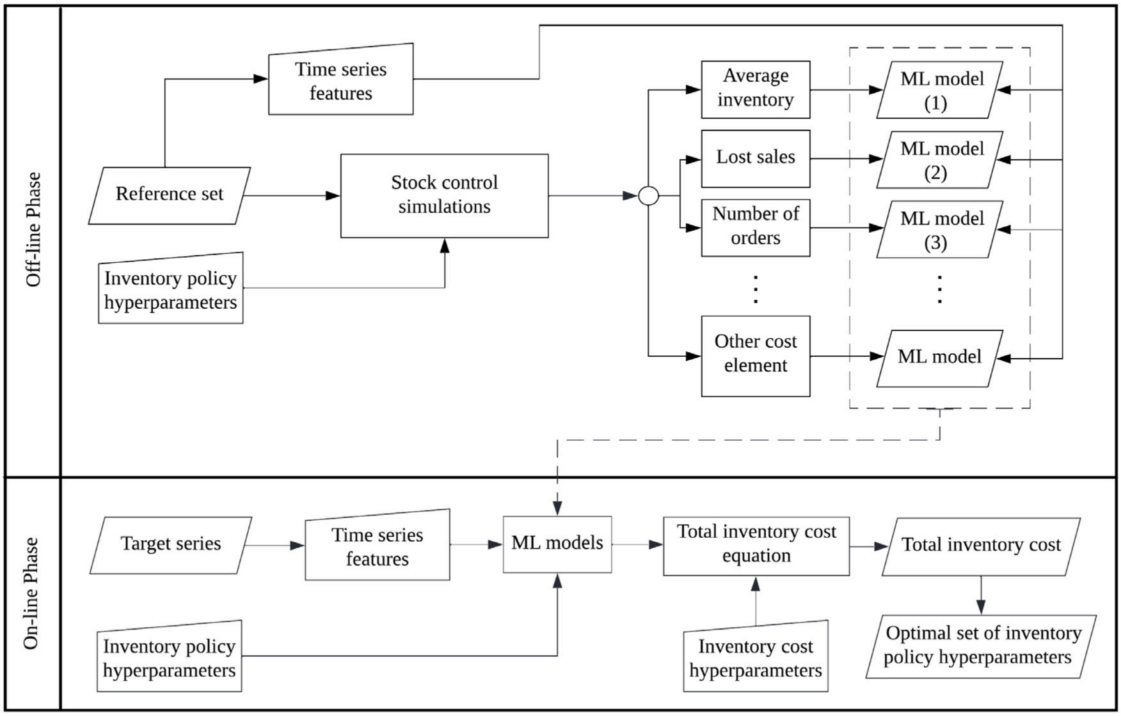

Fig. 2. Flowchart of the proposed framework (MOIC).

3.4. Cost structure

To compute total inventory cost, Eq. (8) is used, assuming a unit holding cost (h) of 1%, a unit shortage cost (b) of 25%,[1] and an ordering cost (k) of \$0.50. Although the choice of these values may affect results, these can be considered representative for grocery retailers on which the present study focuses. Nevertheless, these values are effectively hyperparameters of the proposed framework, meaning that they can be easily adjusted to represent more precisely the inventory control process of any given retailer.

3.5. Inventory cost elements prediction models

Light Gradient Boosting Machine (LightGBM; Ke et al., 2017) models are employed for predicting the inventory cost elements given a particular set of inventory policy hyperparameters as input. LightGBM, which is a special, more efficient variant of gradient boosted trees, has recently been proven particularly successful in various regression applications, outperforming many other state-of-the-art ML methods (Bojer and Meldgaard, 2021). Three separate LightGBM models are trained, each having a different target variable (I, LS, and NO), but the same features (ADI, CV^2, R, L, and TSL).

To enhance the generalization of the models, the target variables are scaled before training by dividing their values by the average demand of the item they correspond to, $\overline{D}$. In addition, key hyperparameters[2] of the LightGBM models, namely the learning rate (`learning_rate`) and the number of leaves (`num_leaves`), are tuned to further improve the accuracy of the forecasts. To do so, the train set is randomly split into a train and a validation set, each consisting of 20% and 80% of the observations originally available for training. Then, a grid search approach is employed to identify the most appropriate value for each hyperparameter from a range of [0.05, 0.5] and [10, 130], respectively. LightGBM is implemented using the `lightgbm` package for R (Ke et al., 2020).

4. Empirical evaluation

4.1. Real data

In order to evaluate the performance of the proposed framework, we consider the data set of the M5 competition which involves the unit sales of 3,049 products sold by Walmart in 10 of its stores (Makridakis et al., 2022). The data are daily and cover a period of 1969 days (from 2011-01-29 to 2016-06-19).

Although the data set originally consists of twelve aggregation levels, grouping the series in a hierarchical fashion based on the location (USA state and store) and the type (category and department) of the items being sold, in this study we focus on the 11_{th} aggregation level (product–state), as shown in Table 1. This was done since the proposed framework is primarily designed for optimizing inventory performance at a warehouse level, meaning that demand should be reported for groups of stores located in the same region. In practice, in this empirical evaluation we assume that the retailer has three warehouses, each located in a different region (the states of California, Texas, and Wisconsin) and fulfilling the orders of particular stores (4 at California and 3 at Texas, and Wisconsin).

From the 9147 series available at the product–state level, 643 were dropped from the data set used in the experiments, resulting in a total of 8504 series. This filtering was done to make sure that all series considered in the simulations involved at least two full calendar years worth of observations, the first year serving the implementation of the off-line phase of MOIC (setting up the framework and training the LightGBM models), while the latter the implementation of the on-line phase (evaluating performance) of the framework.

Note that, in addition to units sales data, the M5 data set provides information about the price of each item (p_i). This information is necessary for computing total inventory cost, as described in Eq. (8), and is therefore included in the conducted experiments.

[1] Given that the present study focuses on grocery retailers, the gross profit margin of Walmart (https://www.statista.com/statistics/269414/gross-profit-margin-of-walmart-worldwide-since-2006/) can serve as a realistic point of reference.

[2] https://lightgbm.readthedocs.io/en/latest/Parameters.html

Table 1
Number of M5 series per aggregation level. The series considered in the present study are displayed in bold.

Level	Description	Aggregation	Series
1	Unit sales of all products, aggregated for all stores/states	Total	1
2	Unit sales of all products, aggregated for each state	State	3
3	Unit sales of all products, aggregated for each store	Store	10
4	Unit sales of all products, aggregated for each category	Category	3
5	Unit sales of all products, aggregated for each department	Department	7
6	Unit sales of all products, aggregated for each state and category	State–category	9
7	Unit sales of all products, aggregated for each state and department	State–department	21
8	Unit sales of all products, aggregated for each store and category	Store–category	30
9	Unit sales of all products, aggregated for each store and department	Store–department	70
10	Unit sales of product i, aggregated for all stores/states	Product	3,049
11	Unit sales of product i, aggregated for each state	Product–state	**9,147**
12	Unit sales of product i, aggregated for each store	Product–store	30,490
Total			42,840

Table 2
Overview of the data sets used in the study. For each data set, the number of series involved and their average lengths, along with the proportion of erratic, lumpy, smooth, and intermittent series in each set are reported.

Data set	Observations	Time series	Erratic	Lumpy	Smooth	Intermittent
Real (M5)	1626	8504	8.01%	12.35%	35.12%	44.52%
Simulated	548	1000	24.40%	25.80%	24.30%	25.50%

4.2. Simulated data

In addition to the M5 data set, we consider a set of simulated data that can serve as an alternative reference set when implementing the off-line phase of the framework. Apart from evaluating the transfer learning capabilities of MOIC, using simulated data as a reference set has another potential advantage compared to real data; the latter type of series do not necessarily express the unconditional demand of the warehouse, being subject to possible stock-outs due to sub-optimized replenishment and delivery delays, among others. As a result, in theory, LightGBM models trained on real data may underestimate demand, resulting in sub-optimal suggestions.

We generate demand data using the simulation approach proposed by Petropoulos et al. (2014). According to this approach, the occurrence of a non-zero demand follows a Bernoulli distribution (Croston, 1972; Syntetos and Boylan, 2001) with $p = \frac{1}{ADI}$, while the demand size follows a negative binomial distribution (Syntetos et al., 2011) with $n = \frac{\mu p}{1-p}$ and $p = \frac{\mu}{CV^2(\mu+1)^2}$, where μ is the average demand. Note that simulated demand size values are increased by one to make sure the demand is strictly positive.

To make sure the simulated data is both diverse and balanced in terms of demand patterns, an equal number of time series are generated for each of the four categories of demand presented in Section 2.1. ADI and CV^2 values were selected randomly from $[1, 30]$ and $[0.1, 2]$, with breaks in 1.32 and 0.49, respectively, while μ was randomly sampled from $[10, 100]$. The resulting data set consists of 1000 daily time series, each of 548 observations (1.5 years), with approximately 25% of them being erratic, lumpy, smooth, and intermittent.

Table 2 summarizes the number and average length of the series included in the M5 and simulated data sets, also indicating the proportion of series that are classified as erratic, lumpy, smooth, and intermittent per case. The distribution of the series across the four categories of demand (ADI and CV^2 values in logarithmic scale) is visualized in more detail in Fig. 3. It is evident that the simulated series are significantly shorter and fewer in numbers compared to the M5 ones, also displaying notable variations in terms of demand patterns. These key differences have been intentionally designed in order to be possible to evaluate the transfer learning capabilities of the proposed framework more objectively and conclude whether the LightGBM models used are effectively generalized to provide accurate results, even in cases where the reference series are not similar with the target ones.

4.3. Experimental setup

In order for the conclusions drawn to be more objective and account for possible trends and seasonal dynamics, we evaluate MOIC using the last year of the M5 data set (on-line phase), retaining the remaining of the M5 historical data or the complete simulated data set for training (off-line phase).

We follow the steps of the algorithm presented in Section 3 to train two different sets of LightGBM model, the first, ML_{Sim}, predicting the inventory cost elements based on the simulated series, while the second, ML_{M5}, based on the M5 series. A representation of the procedure described above is presented in Fig. 4. Taking into account the time series features of the M5 historical data, as well as the selected cost hyperparameters, each set of models (ML_{Sim} and ML_{M5}) conducted several simulations and identified a different set of inventory policy hyperparameters as "optimal" for each M5 series, as presented in Fig. 5.

Fig. 6 presents the feature importance of the three ML_{Sim} and ML_{M5} LightGBM models trained for predicting the inventory cost elements. For the ML_{Sim} set of models we observe that ADI is the most critical forecast variable, followed by CV^2 and R. L and TSL are of much lower importance, accounting for less than 10% of the overall gain. These findings suggest that (i) the inventory cost elements are affected more by the patterns of the demand than the inventory policy hyperparameters used, (ii) variability of demand sizes can significantly affect lost sales and orders, (iii) inventory level is mostly affected by intermittency, and (iv) the review period is mostly helpful for determining the number of orders and inventory level. However, different conclusions can be drawn if we focus on the importance of the features for the ML_{M5} set of models. In this case, some of the inventory policy hyperparameters (e.g. R and L) are of similar or even greater importance than ADI and CV^2, indicating that selecting different values for such hyperparameters can significantly affect inventory cost. These differences demonstrate that models trained on different reference sets capture different data relationships and, as a result, identify different inventory practices as "optimal".

For reasons of comparison, we also consider two benchmarks. The first benchmark refers to the "true optimal" (TO) inventory policy hyperparameters of each M5 series that led to the lowest inventory cost in the historical data. Effectively, this benchmark indicates what the performance of the retailer would be in the following year if they optimized replenishment based on what worked best in the previous year. The second benchmark involves a fixed set of inventory policy hyperparameters (FSH). The selected values of these parameters were

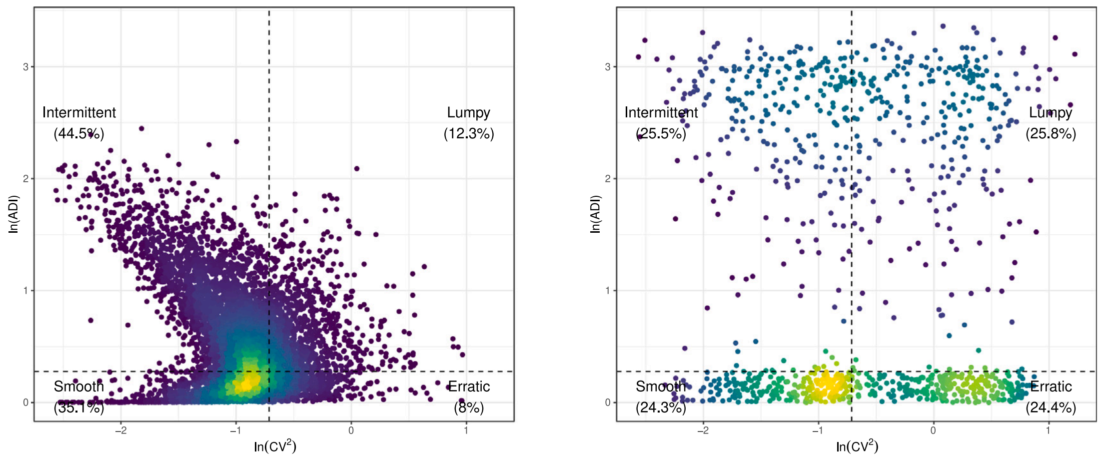

Fig. 3. Demand classification of the real (M5; left) and simulated (right) series based on their intermittency (ADI) and erraticness (CV^2).

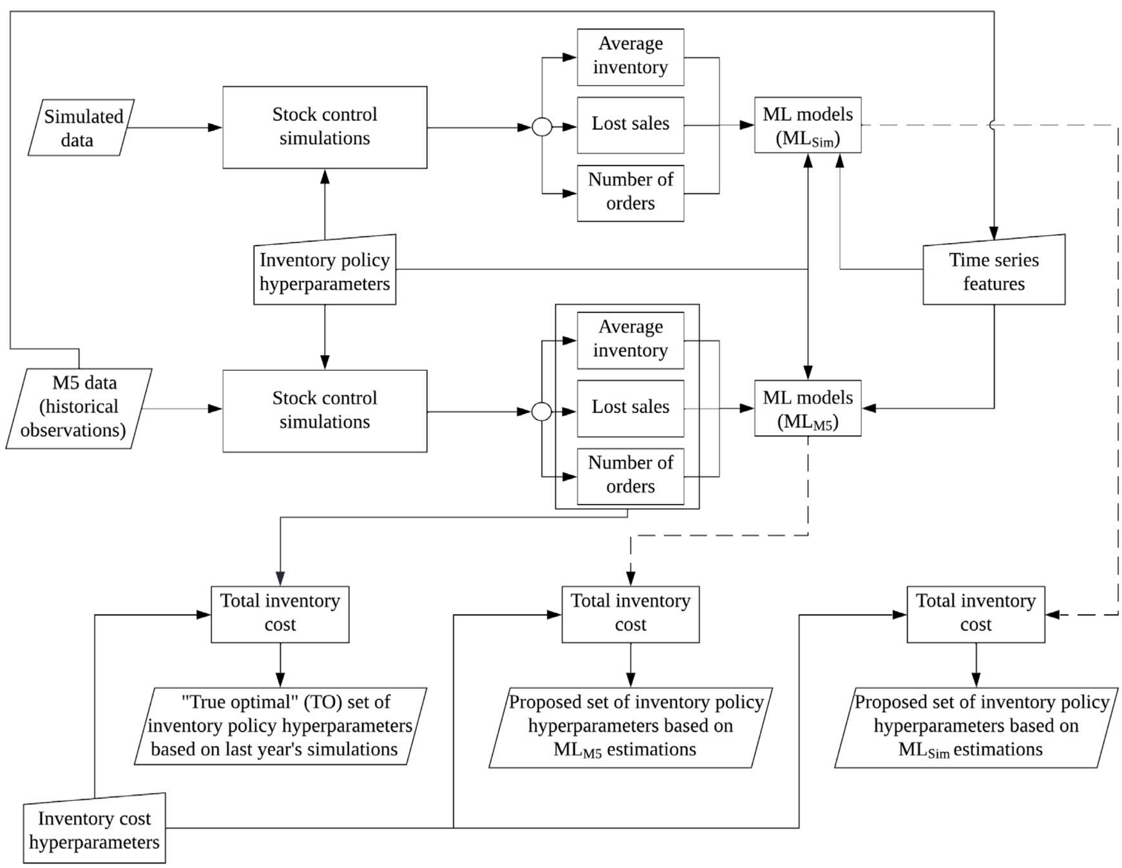

Fig. 4. Flowchart of the experimental setup: off-line phases.

$R = 7$, $L = 3$, and $TSL = 0.95$, thought to be close to the actual average ones used in the industry (according to these values the retailer reviews the inventory once per week, the supplier fulfills the orders, if any, within 3 days, and stock-outs should occur in 5% of the time). Although one may argue that the FSH benchmark is arbitrary parameterized, being possibly far from optimal, as we demonstrate in Appendix B, the selected set of values is actually particularly competitive.

5. Results and discussion

Table 3 summarizes the results of the two MOIC approaches (ML_{Sim} and ML_{M5}) and the benchmarks considered. This includes the realized total inventory cost (C_{Tot}) of each replenishment strategy, as well as its components, i.e. holding cost (C_H), lost sales cost (C_{LS}), and ordering cost (C_O). The average realized service level ($\overline{SL}$) of each strategy is also reported.

As seen, both MOIC approaches outperform the selected benchmarks, both in terms of total cost and service level, with the transfer learning approach reporting the best results. In particular, ML_{Sim} reduces total cost over the TO benchmark by about 24%, while improving service level by 2%. According to the simulations, most of these savings originate from less lost sales and lower holding costs. Moreover, compared to ML_{M5}, ML_{Sim} reduces cost by about 13%, while increasing the service level by 1% (again by reducing lost sales and inventory days). It becomes evident that the transfer learning approach is more effective on average in orchestrating the replenishment, placing enough

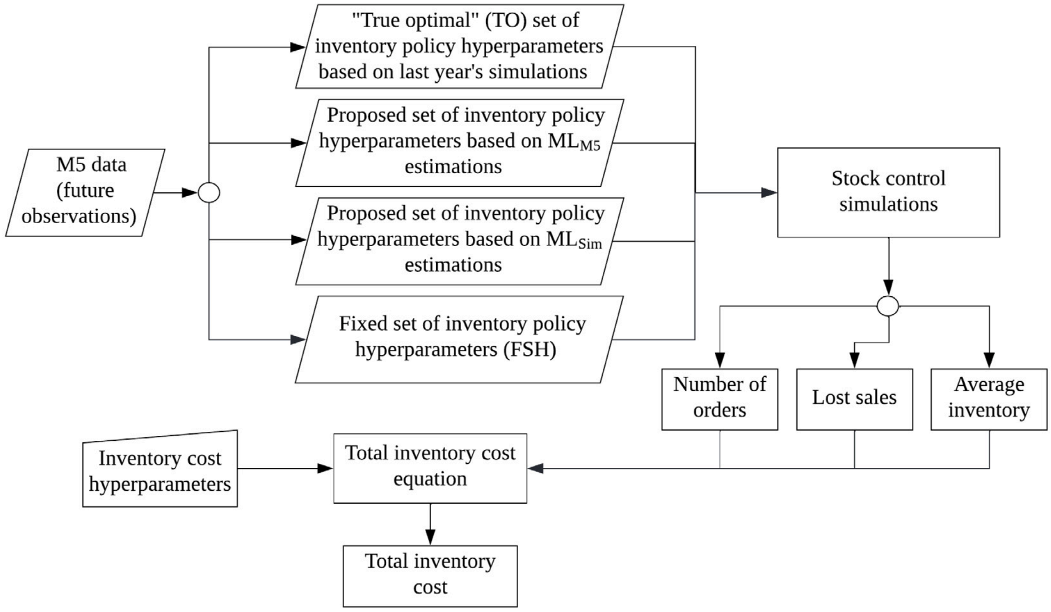

Fig. 5. Flowchart of the experimental setup: on-line phase and evaluation.

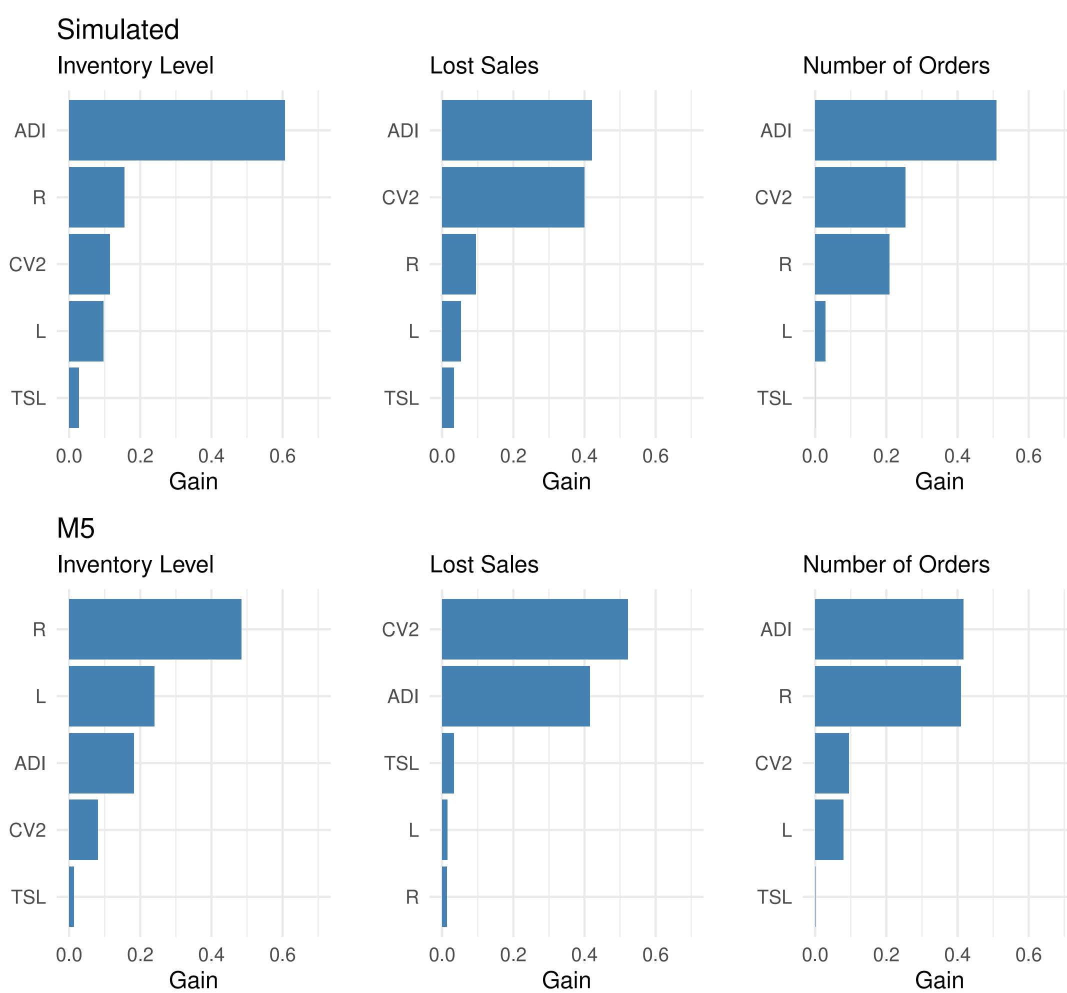

Fig. 6. Feature importance (gain) of the three LightGBM models comprising the ML_{Sim} (top) and ML_{M5} (bottom) set of models.

orders to cover the demand, while not exaggerating. We also find that TO performs similar to the FSH strategy, being slightly better in terms of total cost, but worse in terms of service level. This is because, according to the FSH strategy, orders are placed on a very regular basis, thus decreasing lost sales to some extent, but on the expense of significantly higher ordering costs. Interestingly, although the FSH

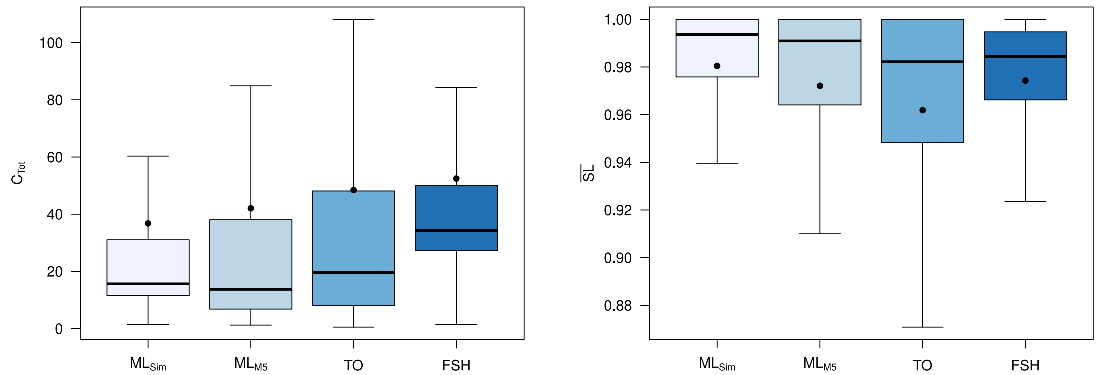

Fig. 7. Inventory performance of the examined replenishment strategies (MOIC approaches and benchmarks). The box-plots represent the distribution of inventory cost (left) and service level (right) across the 8504 series of the M5 data set.

Table 3
Performance of the examined replenishment strategies (MOIC approaches and benchmarks) in terms of inventory cost and service level. The results are reported for all 8504 series of the M5 data set.

Approach	C_H	C_{LS}	C_O	C_{Tot}	$\overline{SL}$
ML_{M5}	36,705	286,658	**34,189**	357,552	0.97
ML_{Sim}	23,035	**214,869**	74,875	**312,779**	**0.98**
TO	39,183	333,522	39,232	411,937	0.96
FSH	**10,820**	246,502	188,850	446,172	0.97

results imply more frequent orders than ML_{Sim}, the latter approach still leads to less lost sales. This observation can be attributed to the demand patterns of the M5 series; due to the uncertainty present when the demand is erratic and/or intermittent, more frequent orders do not necessarily guarantee better inventory performance. This finding is in agreement with the results of Fig. 6, highlighting the importance of ADI and CV^2 in the replenishment process.

We further investigate the performance of the proposed framework by summarizing the results per category of demand, as shown in Table 4. Similarly to the whole data set, ML_{Sim} outperforms all other strategies, both in terms of total inventory cost and realized service level. We also find that the percentage improvements of employing the ML_{Sim} approach over another strategy are greater for the erratic and smooth series, which is expected given that these types of items are moving faster and, as a result, contribute more to the revenue and costs of the retailer. For instance, compared to ML_{M5}, ML_{Sim} reduces cost by about 4.5% when used to control intermittent and lumpy items, and by about 18.5% when used to replenish erratic and smooth items. These improvements rise to about 18% and 26% for the case of the TO strategy, respectively. This finding suggests that MOIC can result in notable improvements, regardless the type of demand patterns the items display.

To investigate the robustness of the examined replenishment strategies on an item basis, Fig. 7 presents the respective distributions of the realized inventory performances across the 8,504 series of the M5 data set. As seen, ML_{Sim} results in narrower distributions of better median values. Moreover, although FSH and ML_{M5} distributions have similar variations, the latter is much better in terms of median performance, highlighting the benefits of the MOIC approach. Similar conclusions can be drawn from Fig. 8 that disaggregates the results both per category of demand and region (warehouse). Drawing from these findings, we conclude that MOIC performance is superior, both on average and across various items and warehouses.

The reason why ML_{Sim} (models trained with simulated data) performs better than ML_{M5} (models trained with the same data being forecast) may be attributed to the fact that, as explained in Section 4.2, the M5 data set represents the realized sales of the products and not their unconditional demand. Given that the pattern of the unconditional demand may be disrupted in practice by stock outs, sub-optimized replenishment, and delivery delays, the train set of the ML_{M5} models can become biased, forcing towards recommendations that underestimate demand. In addition, due to its nature, the M5 data set mostly consists of smooth and intermittent series, meaning that ML_{M5} may provide sub-optimal results for the underrepresented categories of demand (erratic and lumpy series). This is not the case for ML_{Sim} that is trained on a balanced data set in terms of demand pattern categories, considering series with undisrupted demand. To that end, ML_{Sim} models generalize better and lead to better inventory performance.

As explained in Section 3.3, the proposed framework is independent from the inventory policy used for replenishment. To illustrate this feature, the previous experiments were repeated using the (s, S) policy, a special case of the (R, s, S) policy. According to (s, S), the inventory is monitored daily and an order can be placed at any time. This policy is particularly popular among retailers since it is less difficult to optimize and allows for an instant placement of orders. Therefore, when it comes to MOIC, the optimization process of (s, S) is similar to that conducted for (R, s, S), with the exception that the inventory hyperparameters of the former are less in numbers (there is no review period R to optimize). The results of the additional experiments are presented in Table 5. As seen, our key findings, drawn based on the (R, s, S) experiments, are in general agreement with those drawn based on the (s, S) ones. This verifies that the proposed framework can indeed be applied successfully for different inventory policies. Equally important, this property of MOIC allows making fast comparisons between alternative policies, i.e. selecting the most appropriate inventory policy in addition to selecting the optimal inventory hyperparameters per policy. The only notable difference between Tables 4 and 5 is that the improvements of MOIC over the TO benchmark are significantly lower. This is because in the (s, S) policy the inventory hyperparameters (and alternative solutions) are fewer, meaning that the task assigned to the models is much simpler and, as a result, the margin of improvement relatively smaller.

As a final step to our analysis, we investigate the savings that MOIC can offer in terms of computational time over standard, simulation-based replenishment strategies. Given that traditional inventory control optimizations can become particularly intensive, computational cost should not be overlooked, especially when performed for thousands or millions of items on a regular basis (Seaman, 2018). Fig. 9 illustrates the time required for simulating the inventory performance of a given set of 850 items over multiple sets (up to 9,000) of inventory policy and inventory cost hyperparameters, either by applying the respective inventory policy directly on the historical data or by employing the MOIC framework. As seen, when the trials are exhaustive (850 series × 9,000 sets of hyperparameters), standard simulations are complete after 28 days. On the contrary, MOIC estimations are complete in less than 12 h. As a result, approximating inventory performance with MOIC is

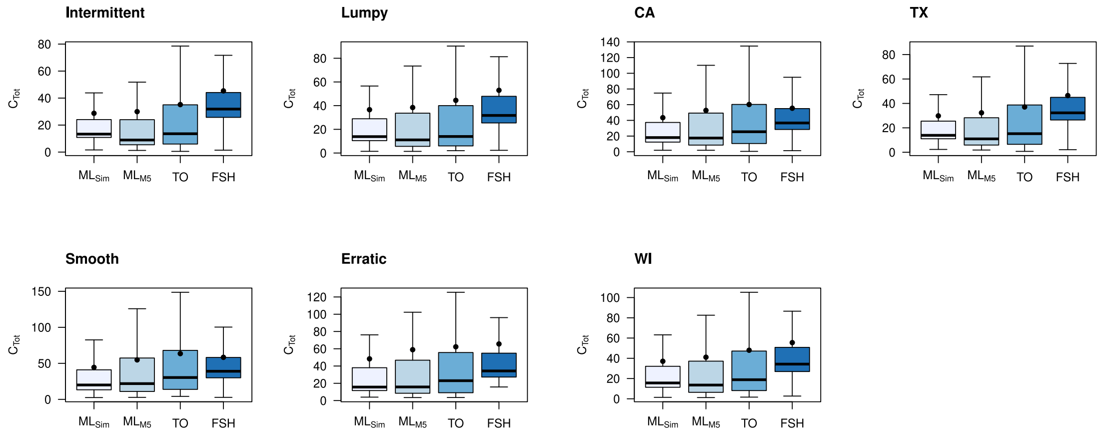

Fig. 8. Inventory performance of the examined replenishment strategies (MOIC approaches and benchmarks). The box-plots represent the distribution of inventory cost per category of demand (left) and region (right).

Table 4
Performance of the examined replenishment strategies (MOIC approaches and benchmarks) in terms of inventory cost and service level. The results are reported for each category of demand separately.

Approach	C_{Tot}				$\overline{SL}$			
	Erratic	Intermittent	Lumpy	Smooth	Erratic	Intermittent	Lumpy	Smooth
ML_{M5}	40,130	113,395	40,336	163,690	0.98	0.97	0.96	0.98
ML_{Sim}	**32,920**	**108,779**	**38,416**	**132,665**	**0.98**	**0.98**	**0.97**	**0.99**
TO	42,415	133,170	46,702	189,649	0.97	0.96	0.95	0.97
FSH	44,655	171,611	55,616	174,289	0.98	0.97	0.96	0.99

Table 5
Performance of the examined replenishment strategies (MOIC approaches and benchmarks) in terms of inventory cost and service level. The results are reported for all 8504 series of the M5 data set using the (s, S) policy.

Approach	C_H	C_{LS}	C_O	C_{Tot}	$\overline{SL}$
ML_{M5}	39,582	438,810	254,818	733,210	0.96
ML_{Sim}	40,242	440,578	**252,227**	**733,047**	0.96
TO	27,138	**364,314**	344,048	735,500	0.96
FSH	**4,611**	421,083	910,814	1,336,508	0.96

about 70 times faster than conducting standard simulations, resulting in significant savings in computational resources.

The reason behind the significant differences reported in terms of computation time between the proposed framework and the simulation method is that the former builds on pre-trained and operationalized models: Once a ML model is trained, inference can be performed particularly fast, effectively outperforming computationally-wise any simulation method of the same number of iterations. This finding points out that once the off-line phase of MOIC is completed, having trained the models with the same or another data set, the time needed to estimate the total inventory cost for any set of inventory policy parameters and time series is significantly less than conducting a simulation, rendering the proposed framework a powerful tool for testing various scenarios or identifying near optimal values.

6. Conclusion

This paper introduced MOIC, a framework that can approximate inventory performance for any given set of items, while considering an inventory policy setup and a cost function of interest. Inspired by the work done in the field of machine learning, the proposed approach exploits transfer learning to relate demand patterns and key inventory policy hyperparameters with total inventory cost and, based on the relationships learned, optimize stock control.

The performance of the proposed framework was evaluated using a large data set of 8504 series that correspond to the demand of ten Walmart stores, located in three USA states. The assessment was performed both in terms of total inventory cost and service level, conducting simulations over a period of one year on a daily basis and making comparisons with indicative benchmarks. The results suggest that when MOIC is used to define an appropriate review period, lead time, and target service level for each item separately, inventory cost can be reduced by up to 24%, while product availability can be increased by 2%. Moreover, we find that these improvements are relatively consistent across individual series of different demand patterns and selling locations. Finally, we show that MOIC suggestions can be produced 70 times faster than using standard inventory simulations, thus saving valuable computational resources.

Future work could focus on further exploiting the multifaceted nature of the proposed framework, allowing for more flexibility and improvements. This includes adjusting MOIC to account for additional inventory policies than (R, s, S), considering more advanced methods to forecast demand, and investigating different machine learning models to approximate inventory level, lost sales, and number of orders. In addition, the framework could be expanded to consider other types of costs, limitations related with batch ordering, as well as possible uncertainties about lead times, transportation, and shortage costs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Sample of series

This appendix provides indicative examples of the real (M5) and simulated series used in the present study. For each category of demand (intermittent, lumpy, smooth, and erratic), a random series was selected per case (see Figs. A.1 and A.2).

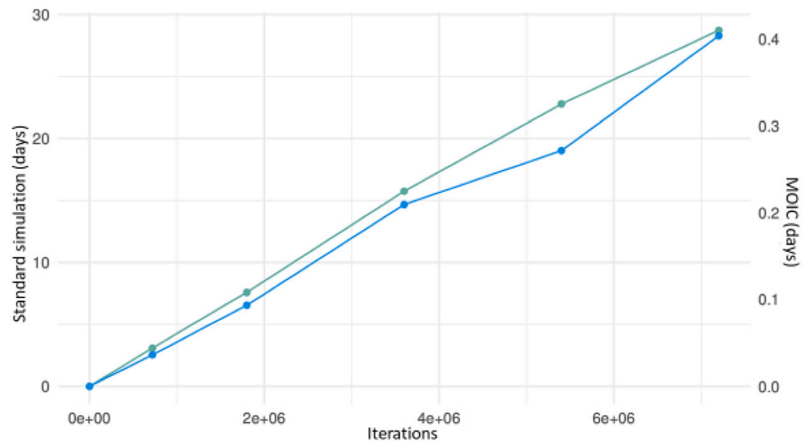

Fig. 9. Time required for simulating inventory performance for 850 series using standard simulations versus the MOIC framework.

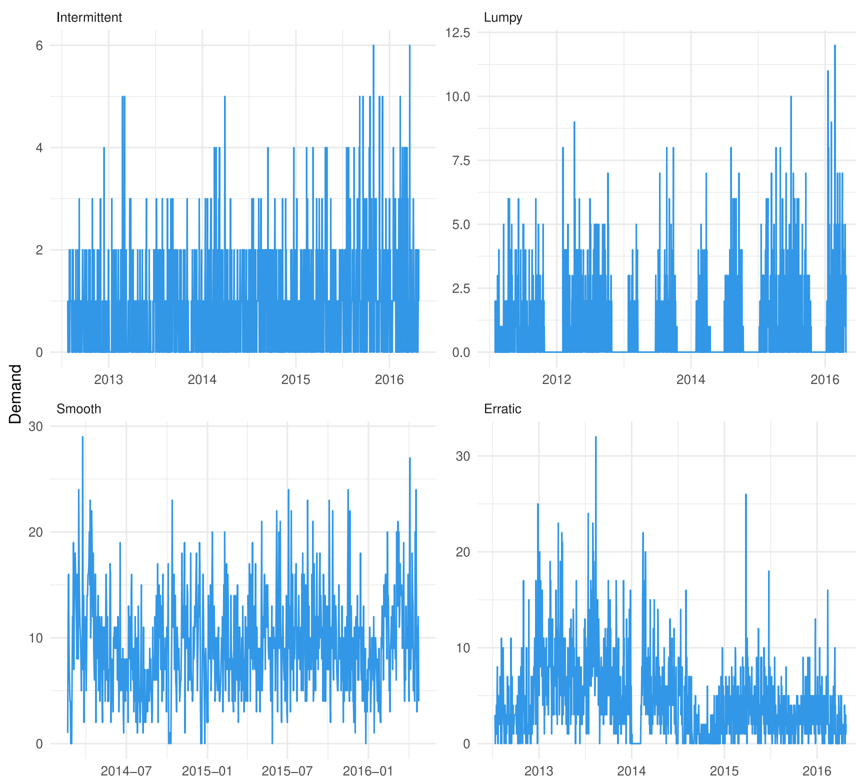

Fig. A.1. Sample of series from the real (M5) data set.

Appendix B. FSH benchmark performance analysis

The box-plots of Fig. B.1 present the distribution of the total inventory cost (C_{Tot}) when variations of the FSH benchmark (the same inventory hyperparameter values are selected for all products) are used for replenishing the complete set of products included in the M5 data set. The sub-figure on the left depicts the results of the experiments for the historical observations of the M5 (year before the one used for evaluation), while the sub-figure on the right for the future observations (year used for evaluation).

The figure on the left clearly indicates that the set of values selected for implementing the FSH benchmark in the present study ($R = 7, L = 3, TSL = 0.95$) is among the top performing ones. Moreover, it demonstrates that the second of the benchmarks used, TO, is capable of effectively identifying the most appropriate set of values for each product separately. In this respect, both benchmarks can be considered as reasonable and competitive.

The figure on the right confirms the observations made above, but it also suggests that, although the best set of values from the previous year (TO) lead to better inventory performance than any fixed set of values for all products (FSH), it does not achieve the lowest cost in the year used for evaluation. This is because, due to changes in the demand patterns, some products may perform better with different sets of values from one year to another. It is also evident that the sets of hyperparameters proposed by the ML models are able to identify these correlations between the demand patterns and the inventory cost elements, resulting in lower total inventory cost than any of the benchmarks.

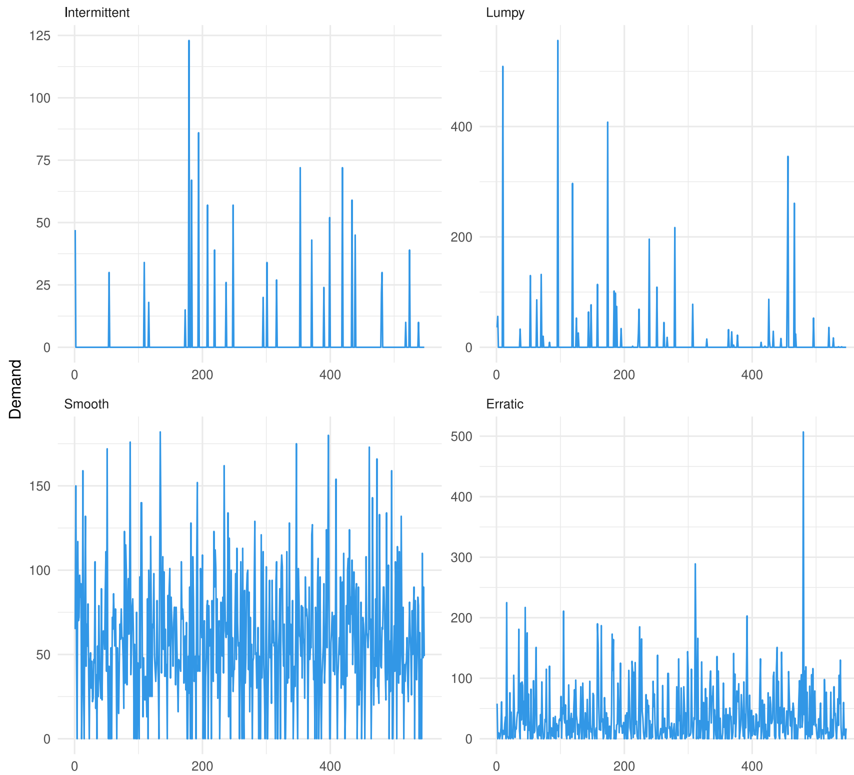

Fig. A.2. Sample of series from the simulated data set.

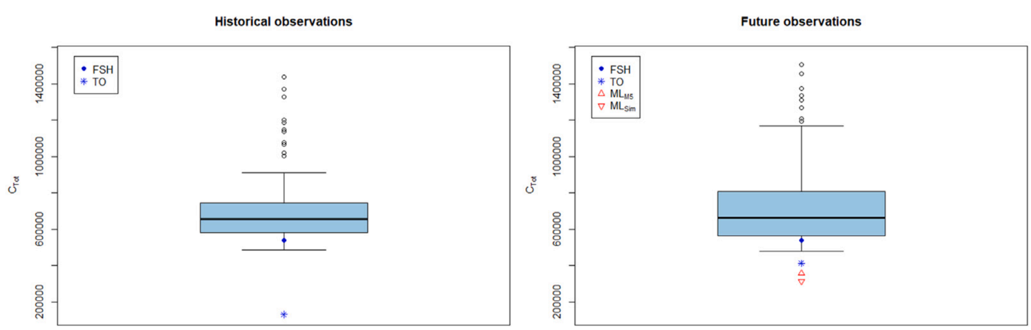

Fig. B.1. Box-plots showing the resulted total inventory cost for various sets of inventory policy hyperparameters (a) on the historical observations (left) (b) on the future observations of the M5 data set (right).

References

Alfares, H.K., Elmorra, H.H., 2005. The distribution-free newsboy problem: Extensions to the shortage penalty case. Int. J. Prod. Econ. 93–94, 465–477, Proceedings of the Twelfth International Symposium on Inventories.

Altay, N., Litteral, L.A., Rudisill, F., 2012. Effects of correlation on intermittent demand forecasting and stock control. Int. J. Prod. Econ. 135 (1), 275–283, Advances in Optimization and Design of Supply Chains.

Anderson, D.R., Sweeney, D.J., Williams, T.A., Camm, J.D., Cochran, J.J., 2018. An Introduction to Management Science: Quantitative Approach. Cengage learning.

Berling, P., Rosling, K., 2005. The effects of financial risks on inventory policy. Manage. Sci. 51 (12), 1804–1815.

Bijvank, M., Vis, I.F., 2012. Lost-sales inventory systems with a service level criterion. European J. Oper. Res. 220 (3), 610–618.

Bojer, C.S., Meldgaard, J.P., 2021. Kaggle forecasting competitions: An overlooked learning opportunity. Int. J. Forecast. 37 (2), 587–603.

Boute, R.N., Gijsbrechts, J., van Jaarsveld, W., Vanvuchelen, N., 2021. Deep reinforcement learning for inventory control: A roadmap. European J. Oper. Res. 298 (2), 401–412.

Boylan, J.E., Syntetos, A.A., Karakostas, G.C., 2008. Classification for forecasting and stock control: a case study. J. Oper. Res. Soc. 59 (4), 473–481.

Croston, J.D., 1972. Forecasting and stock control for intermittent demands. J. Oper. Res. Soc. 23 (3), 289–303.

Eaves, A., Kingsman, B., 2004. Forecasting for the ordering and stock-holding of spare parts. J. Oper. Res. Soc. 55, 431–437.

Federgruen, A., Zheng, Y.-S., 1992. An efficient algorithm for computing an optimal (r, Q) policy in continuous review stochastic inventory systems. Oper. Res. 40 (4), 808–813.

Federgruen, A., Zipkin, P., 1984. An efficient algorithm for computing optimal (s, S) policies. Oper. Res. 32 (6), 1268–1285.

Gardner, Jr., E.S., 1985. Exponential smoothing: The state of the art. J. Forecast. 4 (1), 1–28.

Gautam, Y., 2021. Transfer Learning for COVID-19 cases and deaths forecast using LSTM network. ISA Trans..

Goltsos, T.E., Syntetos, A.A., Glock, C.H., Ioannou, G., 2022. Inventory – forecasting: Mind the gap. European J. Oper. Res. 299 (2), 397–419.

Greasley, A., 2013. Operations Management, third ed. John Wiley & Sons Limited, England.

Harris, F.W., 1913. How Many Parts To Make at Once. pp. 135–136. 152, Factory, The Magazine of Management 10.

Heath, D.C., Jackson, P.L., 1994. Modeling the evolution of demand forecasts ITH application to safety stock analysis in production/distribution systems. IIE Trans. 26 (3), 17–30.

Hemalatha, S., Annadurai, K., 2020. Inventory models involving lead time crashing cost as an exponential function with ordering cost reduction dependent on lead time. Mater. Today: Proc..

Hill, R.M., Johansen, S.G., 2006. Optimal and near-optimal policies for lost sales inventory models with at most one replenishment order outstanding. European J. Oper. Res. 169 (1), 111–132.

Inprasit, T., Tanachutiwat, S., 2018. Reordering point determination using machine learning technique for inventory management. In: 2018 International Conference on Engineering, Applied Sciences, and Technology. ICEAST, IEEE, pp. 1–4.

Johnston, F.R., Boylan, J.E., 1996. Forecasting for items with intermittent demand. J. Oper. Res. Soc. 47 (1), 113–121.

Johnston, F.R., Shale, E.A., Kapoor, S., True, R., Sheth, A., 2011. Breadth of range and depth of stock: forecasting and inventory management at Euro Car Parts Ltd. J. Oper. Res. Soc. 62 (3), 433–441.

Karlin, S., Scarf, H., 1958. Inventory Models of the Arrow-Harris-Marschak Type with Time Lag. Stanford University Press, Stanford, Ca.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 30, Curran Associates, Inc..

Ke, G., Soukhavong, D., Lamb, J., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2020. Lightgbm: Light gradient boosting machine. R package version 3.1.1.

Lagodimos, A., Christou, I., Skouri, K., 2012. Computing globally optimal (s,S,T) inventory policies. Omega 40 (5), 660–671.

Lambert, D.M., Stock, J.R., 1993. Strategic Logistics Management. Vol. 3, Irwin Homewood, IL.

Liao, Y., Banerjee, A., Yan, C., 2011. A distribution-free newsvendor model with balking and lost sales penalty. Int. J. Prod. Econ. 133 (1), 224–227.

Lolli, F., Balugani, E., Ishizaka, A., Gamberini, R., Rimini, B., Regattieri, A., 2019. Machine learning for multi-criteria inventory classification applied to intermittent demand. Prod. Plan. Control 30 (1), 76–89.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022. M5 accuracy competition: Results, findings, and conclusions. Int. J. Forecast. 38 (4), 1346–1364.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022. The M5 competition: Background, organization, and implementation. Int. J. Forecast. 38 (4), 1325–1336.

Mišić, V.V., Perakis, G., 2020. Data analytics in operations management: A review. Manuf. Serv. Oper. Manag. 22 (1), 158–169.

Namir, K., Labriji, H., et al., 2022. Decision support tool for dynamic inventory management using machine learning, time series and combinatorial optimization. Procedia Comput. Sci. 198, 423–428.

Petropoulos, F., Makridakis, S., Assimakopoulos, V., Nikolopoulos, K., 2014. 'Horses for courses' in demand forecasting. European J. Oper. Res. 237 (1), 152–163.

Romer, D., 1996. Advanced Macroeconomics. McMGrawMHill Companies Inc. Singapore.

Scarf, H., 1960. The Optimality of (S, S) Policies in the Dynamic Inventory Problem. Mathematical Methods in the Social Sciences. Stanford University Press, Stanford, Ca.

Seaman, B., 2018. Considerations of a retail forecasting practitioner. Int. J. Forecast. 34 (4), 822–829.

Shang, K.H., Zhou, S.X., 2010. Optimal and heuristic echelon (r, nQ, T) policies in serial inventory systems with fixed costs. Oper. Res. 58 (2), 414–427.

Silver, E., Pyke, D., Peterson, R., 1998. Inventory Management and Production Scheduling, third ed. Wiley, New York.

Simchi-Levi, D., 2014. OM forum—OM research: From problem-driven to data-driven research. Manuf. Serv. Oper. Manag. 16 (1), 2–10.

Spiliotis, E., Makridakis, S., Semenoglou, A.-A., Assimakopoulos, V., 2020. Comparison of statistical and machine learning methods for daily SKU demand forecasting. Oper. Res.: Int. J. 1–25.

Strijbosch, L., Moors, J., 2002. Simulating an (R,s,S) inventory system. Technical Report, Econometrics, Pagination: 13.

Sustrova, T., 2016. An artificial neural network model for a wholesale company's order-cycle management. Int. J. Eng. Bus. Manag. 8, 2.

Syntetos, A., Babai, M., Davies, J., Stephenson, D., 2010. Forecasting and stock control: A study in a wholesaling context. Int. J. Prod. Econ. 127 (1), 103–111.

Syntetos, A.A., Babai, M.Z., Lengu, D., Altay, N., 2011. Distributional assumptions for parametric forecasting of intermittent demand. In: Altay, N., Litteral, L.A. (Eds.), Service Parts Management: Demand Forecasting and Inventory Control. Springer London, London, pp. 31–52.

Syntetos, A., Boylan, J., 2001. On the bias of intermittent demand estimates. Int. J. Prod. Econ. 71 (1), 457–466.

Syntetos, A.A., Boylan, J.E., 2005. The accuracy of intermittent demand estimates. Int. J. Forecast. 21 (2), 303–314.

Syntetos, A., Boylan, J., 2006. On the stock control performance of intermittent demand estimators. Int. J. Prod. Econ. 103, 36–47.

Syntetos, A.A., Boylan, J.E., Croston, J.D., 2005. On the categorization of demand patterns. J. Oper. Res. Soc. 56 (5), 495–503.

Theodorou, E., Wang, S., Kang, Y., Spiliotis, E., Makridakis, S., Assimakopoulos, V., 2022. Exploring the representativeness of the M5 competition data. Int. J. Forecast. 38 (4), 1500–1506.

Tsai, S.C., Chen, S.T., 2017. A simulation-based multi-objective optimization framework: A case study on inventory management. Omega 70, 148–159.

Wellens, A.P., Udenio, M., Boute, R.N., 2022. Transfer learning for hierarchical forecasting: Reducing computational efforts of M5 winning methods. Int. J. Forecast. 38 (4), 1482–1491.

Williams, T., 1984. Stock control with sporadic and slow-moving demand. J. Oper. Res. Soc. 35, 939–948.

Yang, C.-T., 2014. An inventory model with both stock-dependent demand rate and stock-dependent holding cost rate. Int. J. Prod. Econ. 155, 214–221.

Ye, R., Dai, Q., 2021. Implementing transfer learning across different datasets for time series forecasting. Pattern Recognit. 109, 107617.

Ye, R., Dai, Q., 2022. A relationship-aligned transfer learning algorithm for time series forecasting. Inform. Sci. 593, 17–34.

Zimmermann, M., Frejinger, E., 2020. A tutorial on recursive models for analyzing and predicting path choice behavior. EURO J. Transp. Logist. 9 (2), 100004.