**Les Cahiers du GERAD** ISSN: 0711–2440

# A Cost Focused Machine Learning framework for replenishment decisions under transportation cost uncertainty

D. Müllerklein, P. Fontaine, J. Ortmann

G–2024–04

January 2024



La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

**GERAD** HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

**Tél. : 514 340-6053**
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

# A Cost Focused Machine Learning framework for replenishment decisions under transportation cost uncertainty

**Daniel Müllerklein** [a]
**Pirmin Fontaine** [a]
**Janosch Ortmann** [b, c, d]

[a] *School of Management, Catholic University of Eichstaett-Ingolstadt, Germany*

[b] *GERAD, Montréal (QC), Canada, H3T 1J4*

[c] *Centre de recherches mathématiques (CRM), Montréal (QC), Canada, H3T 1J4*

[d] *Département d'analytique, opérations et TI, Université du Québec à Montréal, Canada, H2X 3X2*

daniel.muellerklein@ku.de
pirmin.fontaine@ku.de
ortmann.janosch@uqam.ca

**January 2024**
**Les Cahiers du GERAD**
**G–2024–04**

**Abstract :** Determining optimal inventory replenishment decisions requires balancing the costs of excess inventory with shortage risks. While demand uncertainty has been the focus of stochastic inventory modeling, the effects of transportation cost uncertainty are poorly understood. In practice, transportation modes are prone to disruptions that result in stops and cost increases. While historical disruption data is available, it is difficult for practitioners to understand how replenishment orders must be adjusted. To overcome this gap, we combine mathematical optimization with machine learning to predict cost-optimal replenishment orders using only historical data. The problem is modeled as Stochastic Inventory Routing Problem with Direct Deliveries (SIRPDD) that minimizes total expected costs. With perfect information, optimal decisions are generated as labels for the supervised learning using features from inventory control and disruption-related information. We propose a new Cost Focused Machine Learning (CFML) framework that optimizes the costs of applying replenishment policies within hyperparameter tuning instead of the prediction score of the individual decisions. To handle the resulting computational complexity, we develop a genetic algorithm. We present a case study for the SIRPDD with transportation cost uncertainty. This case, based on a chemical company on the river Rhine, considers two suppliers, different lead times, order sizes, and direct deliveries. Relevant features include the inventory position, historical water level, their trends, and predictions. We show that our CFML can reduce costs by 20% compared to the $(s, Q)$-reorder policy, which represents the industry standard, and 18% compared to classical machine learning frameworks.

**Keywords:** Inventory management, production, artificial intelligence

**Résumé :** Le problème des décisions optimales de réapprovisionnement des stocks vise à équilibrer les coûts des stocks excédentaires et les risques de pénurie. Dans la modélisation stochastique des stocks, les effets de l'incertitude de la demande sont mieux compris que ceux de l'incertitude des coûts de transport. Bien que des données historiques sur les perturbations soient disponibles, il est difficile pour les praticiens de comprendre comment les commandes de réapprovisionnement doivent être ajustées. Pour combler cette lacune, nous combinons l'optimisation mathématique avec l'apprentissage automatique afin de prédire les commandes de réapprovisionnement optimales en utilisant uniquement des données historiques. Le problème est modélisé comme un problème d'acheminement stochastique des stocks avec livraisons directes (SIRPDD), dont l'objectif est de minimiser les coûts totaux attendus. Les décisions optimales sous l'hypothèse de l'information parfaite sont générées en tant qu'étiquettes de l'apprentissage supervisé. Les variables prédictives du modèle sont les caractéristiques du contrôle des stocks et les informations liées aux perturbations. Nous proposons une nouvelle approche que nous appelons l'apprentissage axé sur les coûts (CFML). Notre approche modifie les hyperparamètres selon les coûts d'application de la politique de réapprovisionnement plutôt que le score de prédiction. Pour gérer la complexité informatique qui en résulte, nous développons une version de l'algorithme génétique. Nous présentons une étude de cas pour le SIRPDD sous incertitude des coûts de transport. Ce cas, basé sur une entreprise chimique située au bord du Rhin, prend en compte deux fournisseurs, des délais d'exécution différents, des tailles de commande différentes et des livraisons directes. Les caractéristiques pertinentes comprennent la position des stocks, l'historique des niveaux d'eau, leurs tendances et les prévisions. Alors que les approches d'apprentissage traditionnelles aboutissent à des politiques inefficaces, nous montrons que notre CFML peut réduire les coûts de 20 % par rapport à la politique classique de commande $(s, Q)$ et de 18 % par rapport aux cadres d'apprentissage automatique.

**Mots clés :** Gestion des inventaires, gestion de la production, intelligence artificielle

**Acknowledgements:** We thank the Federal Institute of Hydrology (BfG) for providing the historical water level and forecasting data that supported the analysis. The third author is partially supported by an NSERC Discovery Grant.

# 1 Introduction

Companies worldwide seek a balance between risks and cost-efficiency in their supply chains. Due to the various disruptions in recent years, many companies are undertaking efforts to re-design their supply chains to increase resilience. Following this objective, increasing inventory and sourcing from multiple suppliers are two key levers (Hosseini et al., 2019). In practice, this requires adjusting the policies that set rules for inventory replenishment. Due to their ease of implementation, these policies are industry standard. Determining a cost-optimal inventory replenishment policy, however, that defines when to order, how many units, from which supplier is a challenging problem for decision makers. Historically, extensive efforts have been made to identify the optimal replenishment policy under uncertainty, mainly focusing on demand uncertainty with known stochastic distributions.

However, uncertainties concerning recurring transportation disruptions driven by man-made (e.g., strikes) or natural disruptions (e.g., hurricanes) affect transportation costs and thus the inventory replenishment decisions on a regular level (Chen et al., 2012). Due to the increase in extreme weather events, particularly global inland waterway transport disruptions gained growing attention as low- and high-water-level situations affect the shipment carrier's transport capacities and abilities. This specifically means that, based on the actual water level at the day of a potential shipment, they enforce contractual surcharges on top of their standard prices or even stop their service completely. For example, in 2018, the low water levels at the river Rhine forced a stop of all transportation through the river for 132 days, resulting in production stops due to material shortages for many companies. In 2023, container carriers, including Maersk and Hapag-Lloyd, implemented a low water surcharge between 100-150 USD per Twenty-Foot Equivalent Unit (TEU), a container equivalent, on all cargo transported through the St. Lawrence River in Canada. In addition, low water levels on the Panama Canal forced vessels to lighten their loads. As a result, Hapag Lloyd announced a 500 USD surcharge per TEU on all cargo between Asia and the US East Coast shipped through the canal. Due to climate change, many experts expect an increased frequency of these high and low water events on major waterways in the future (Koetse and Rietveld, 2009). While these surcharges are of high economic importance, the disruptions that drive the future surcharges are uncertain and difficult to predict in advance.

As a result, a decision maker needs to anticipate the relevant transportation costs in delivery time periods in the future as surcharges are determined upon the water level during transport and not upon order. The longer the lead time between ordering and the arrival at the critical transportation mode, the higher the corresponding uncertainty. In addition, practitioners are less interested in a single optimal solution but in decision rules that can help determine an inventory replenishment policy that performs near-optimal under uncertainty. They need to understand which attributes of the problem drive the replenishment decision and how such rules can be extracted from historical data to transfer findings to existing control systems. While an increasing amount of data becomes available, there is still a lack of understanding on how to apply these data-driven inventory management approaches in both practice and academia. In the past, these problems were solved in a two-step framework by estimating random variables and then incorporating these estimations into policy building. Estimating these random variables, however, is a challenging problem itself. Recently, Qi et al. (2023) proposed a data-driven, end-to-end framework to output replenishment quantity decisions under demand and lead time uncertainty directly from input data.

Motivated by the uncertain transportation costs that decision makers can face due to surcharges in waterway transport, we model the problem as SIRPDD (Coelho et al., 2014). This special case of inventory routing takes direct deliveries into account, which is highly relevant in cases where storage capacities and demands of customers are large relative to vehicle capacity, and thus it is optimal to deliver full-truck loads (Gallego and Simchi-Levi, 1990). Delivering full-truck loads is particularly relevant where inventory holding costs are low compared to transport costs. Both holds true for the process industry, which majorly uses inland shipping. We particularly extend the standard problem setting by stochastic transportation costs on arcs connecting the suppliers with the customer location.

Thus, at the time of the replenishment decision, the total procurement cost as the sum of transportation and unit costs are uncertain.

While the introduced multi-stage stochastic programming formulation could be solved in theory, it is clear that obtaining the scenario tree in a data-driven way as well as solving the problem is extremely challenging. Instead, we want to predict future decisions that perform cost-optimal in a data-driven way through a combination of mathematical optimization with machine learning. Thus, we propose a new CFML framework for the inventory replenishment problem with multiple suppliers and transportation cost uncertainty to decide when to source, which quantity, from which supplier. This approach builds on the idea that a replenishment policy can be learned from historical optimal decisions without estimating stochastic assumptions. Within the CFML, we first create labels by solving the Inventory Routing Problem with Direct Deliveries (IRPDD) assuming perfect information. Then, for each order size and each supplier, we train a supervised machine learning algorithm on the labels. In contrast to classical machine learning-optimization frameworks that optimize for classifiers' individual prediction performance, we optimize hyperparameters to directly minimize the costs of applying the resulting replenishment policy. This evaluation is critical as otherwise, inter-dependencies between the individual classifiers are neglected. To overcome the rise in computational complexity, we develop a genetic algorithm with the total costs of applying the predicted replenishment orders as fitness function.

Our main contributions are as follows: (1) We introduce a multi-stage stochastic program for a new type of the SIRPDD with transportation cost uncertainty at every decision stage. The replenishment problem considers multiple suppliers, lead time differences between suppliers, and different transportation quantities. (2) We propose our new CFML framework that directly minimizes the total costs, as defined in the objective function of the SIRPDD, within the validation step of hyperparameter tuning. We show the value of our new framework for data-driven replenishment decisions under uncertainty with multiple suppliers and lead time differences. To handle the rise in computational complexity compared to traditional hyperparameter tuning approaches, we develop a genetic algorithm that incorporates the objective of the SIRPDD as fitness function. (3) We present a case study and compare the CFML against traditional optimization-and-machine learning frameworks and replenishment policies from the literature. Particularly, we show that traditional hyperparameter tuning that maximizes prediction performance leads to inefficient and unstable policies. (4) Via the case study, we derive managerial insights on the data-driven inventory replenishment with multiple suppliers and cost uncertainty, leveraging public databases for waterway transport disruption uncertainty.

This paper is structured as follows. Relevant literature is discussed in Section 2, and the problem is formalized in Section 3. Section 4 presents the SIRPDD with transportation cost uncertainty. The CFML framework is outlined in Section 5. The results of a case motivated by a real-life example are discussed in Section 6. Section 7 summarizes and concludes.

# 2 Literature review

Section 2.1 reviews literature in inventory management under supply uncertainty with a focus on replenishment policies for multiple suppliers. We then shift to data-driven approaches in inventory management in Section 2.2. Lastly, Section 2.3 reviews the use of machine learning in supply chain resilience as a related field, and Section 2.4 summarizes research opportunities.

## 2.1 Inventory management under supply uncertainty and multiple suppliers

Inventory management aims to optimize the holding of stocks along the supply chain to minimize costs while fulfilling customer demand. Over time, a variety of standard problems have been formalized for which mathematically developed decision rules exist that determine when to order, from which supplier, and which quantity. We refer to Silver (1981) for an overview. Despite the early focus on stochastic problem settings (Karlin, 1960), in which models on stochastic demand outweigh those on deterministic demand (Williams and Tokar, 2008), analyzing situations with stochastic supply was not of major attention until the early 2000s (Güllü et al., 1999). In inventory routing, supply uncertainty even gained attention only recently (Alvarez et al., 2021). We summarize the literature concerning supply uncertainty based on its uncertainty characteristics, namely, lead time, quantity, and cost uncertainty. In addition, we distinguish between single and multiple supplier settings as multi-sourcing is a lever for supply-side disruption uncertainty. We recommend the recent overview of Svoboda et al. (2021) on inventory replenishment models with multiple suppliers.

*Lead time* uncertainty analyzes situations in which the replenishment of inventory from a supplier deviates from the contractually agreed time. While the area of lead time uncertainty has been discussed early on in respective literature (Whybark and Williams, 1976; Schmitt, 1984), it is still of relevance today. For example, Chopra et al. (2004) analyze the impact of lead time uncertainty on necessary safety stocks for different service level requirements. Kouvelis and Li (2008) analyze imperfect information on lead times from a primary supplier where a flexible backup supplier can be used as emergency response.

The uncertainty of the *delivery quantity* is relevant in problem settings where only a limited number of suppliers is available, and suppliers face a disruption risk. Thus, inventory can help to reduce the dependency risk on these suppliers. To combine demand uncertainty with the uncertainty of delivery quantity received, Bollapragada and Rao (2006) proposed a heuristic to solve the capacitated inventory system with both demand and supply uncertainty. Iakovou et al. (2010) captured the trade-off between inventory holding and disruption risks at the supplier capacities in a single period stochastic inventory decision-making model extending the classic newsvendor analysis. Similarly, Saghafian and Oyen (2012) analyzed the value of flexible backup suppliers under disruption risks using a newsvendor analysis with recourse. For multiple supplier cases with equal lead times, Chen et al. (2012) propose optimal replenishment decisions in a closed form, studying the problem of a regular supplier that faces disruption uncertainty and a reliable backup supplier at a price premium. They further highlight the need to consider different lead times for suppliers, however, requiring multiple dimensional state spaces to model the system dynamics.

Lastly, *procurement as well as transportation cost* uncertainty is highly relevant in the areas of commodities and fluctuating spot prices in which inventory plays a significant role in risk hedging. One of the first to analyze the impact of fluctuating costs for a single item and multi-period model was Kalymon (1971), assuming a known Markovian stochastic process for the description of future costs. More recently, Xiao et al. (2015) analyze a periodic review joint pricing and inventory control model facing both stochastic demand and fluctuating procurement costs in a dual-sourcing strategy. As one of the rare data-driven approaches, Xiong et al. (2022) use a robust optimization approach for the periodic-review dual sourcing inventory replenishment management in the presence of purchase price and demand uncertainty. The data-driven results lead to some deviating findings from traditional approaches assuming complete distributional information. However, only a robust rolling-horizon model was used due to limited historical data.

## 2.2 Data-driven inventory replenishment

Historical data is available in many companies to develop distribution-free inventory control approaches. Still, most literature (see Section 2.1) relies on distributional assumptions, weakening the accuracy of models at the expense of simplicity (Svoboda et al., 2021). To increase their accuracy in practical applications, data-driven uncertainty modeling is gaining increasing attention. We recommend the overview of Mišić and Perakis (2020) on data-driven approaches in inventory management and summarize relevant works in the following. Priore et al. (2019) use machine learning to choose between four different replenishment policies for a single product in a three-echelon supply chain assuming periodic-review inventory policies and achieve an accuracy above 80% in the prediction of the ideal replenishment rule for the specific supply chain scenario simulated. Neghab et al. (2022) propose an algorithm based on integrating optimization, neural networks, and hidden Markov models for the single-period newsvendor problem with demand that depends on observable features rather than a known demand distribution. Bertsimas et al. (2023) proposed a new approach for adressing multi-stage stochastic problems with unknown distributions where uncertainty is correlated across time by solving a robust optimization problem and providing asymptotic optimality guarantees. They conduct numerical experiments with two single product, single supplier, and stochastic inventory settings where the optimal replenishment quantity is to be determined at each given decision stage. Recently, Qi et al. (2023) proposed a data-driven end-to-end framework for the multi-period decision on inventory replenishment quantities under uncertain demand and lead times using deep-learning. In contrast to this work, they assume a single supplier, fixed replenishment cycles, and conduct the hyperparameter tuning within the validation with the objective of minimizing the difference between predicted and optimal replenishment quantities.

### 2.3 Increasing supply chain resilience with machine learning

While traditional supply chain risk management strategies cannot deal with supply chain disruptions as unexpected events (Pettit et al., 2010), recent focus has shifted towards the usage of big data and advanced analytics for risk prediction and the selection of resilience strategies (Baryannis et al., 2019b; Ivanov et al., 2019). Kosasih and Brintrup (2021) use graph neural networks to detect potential links within a supply chain to increase supply chain visibility to counteract disruptions that occur within the network. Following the objective of increasing transparency, Baryannis et al. (2019a) test different machine learning techniques for risk prediction in combination with expert estimates. Similarly, Brintrup et al. (2020) use a three phase machine learning approach to predict supply-side disruptions at the suppliers with an accuracy of 80%. They highlight the need of incorporating external data sources to increase prediction accuracy and discuss the challenge of class imbalance for disruption-related predictions. Besides, various approaches emerged as a result of the COVID-19 pandemic. For example, Nikolopoulos et al. (2021) use deep-learning models based on google trends and simulating governmental decisions to predict supply chain disruptions while Bassiouni et al. (2023) applied several deep-learning approaches to predict shipment risks that link manufacturers and customers under potential COVID-19 imposed restrictions. Overall, related literature has focused on the ability of disruption predictions with initial success, however, this ability has not been incorporated in corresponding decision models and support systems.

### 2.4 Summary of research opportunities

Operations management research, in general, but particularly research in inventory management usually simplifies their models by assuming distributions for uncertain parameters thereby weakening the accuracy of models (Chen et al., 2023). As a result, there is still a need for data-driven approaches in inventory management (Svoboda et al., 2021). There is especially a need for such approaches that can provide explainable and interpretable results for the acceptance of decision makers (Mišić and Perakis, 2020). Practitioners require decision rules for inventory replenishment that are interpretable by inventory planners and provide adequate decision support. Lastly, there is an ongoing need for inventory replenishment models with supply uncertainty. Specifically, the area of cost uncertainty has mainly been covered through newsvendor approaches that reduce the dimensionality of the replenishment problem to a pure decision on quantity; neglecting the need to decide on when to order from which supplier in a multiperiod setting where past decisions influence the future replenishments.

## 3 Problem formulation

A single product is distributed with direct deliveries from multiple suppliers to a single customer location with unlimited transportation capacity over a finite horizon of discrete time periods. Over the time periods, each direct delivery is done through a specific transportation mode that differs by supplier in terms of transportation costs and delivery times. The single customer location can store inventory to satisfy its demands, whereas unlimited supply capacity is available from suppliers within the delivery time. The customer has a known inventory storage capacity. At the beginning of each period, deliveries from suppliers arrive that were ordered at delivery time periods in the past. Then, a customer's demand is fulfilled, which is known and constant as determined by a production plan. In case the demand exceeds the available inventory, shortages occur. After the fulfillment of customer demand, replenishment orders are placed. Lastly, at the end of each time period an inventory quantity remains that forms the starting inventory of the next time period. Inventory holding and shortage costs are charged at the end of the period, while transportation costs are determined at the time period of arrival. Figure 1 visualizes the overall problem. Here, in time period $t$ the decision maker has two suppliers as options to place an order that will either arrive in $t + 3$ or $t + 5$. Due to the potential occurrence of disruptions, the corresponding costs of both options are still uncertain in $t$ and will only become known upon arrival, thus in $t + 3$ and $t + 5$.

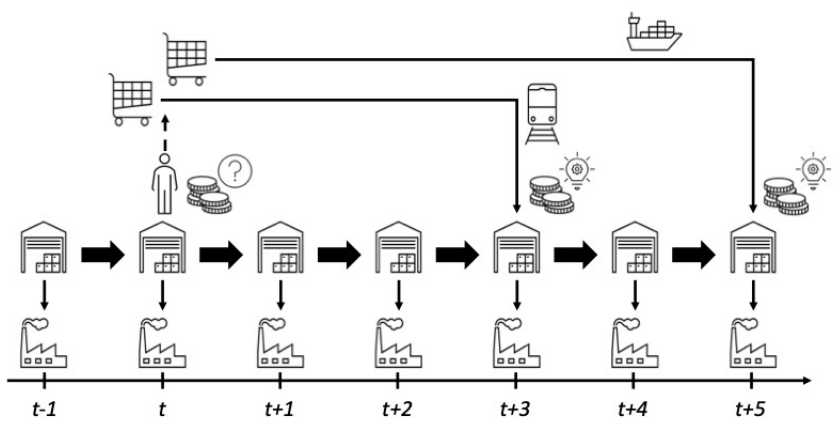


**Figure 1: Inventory replenishment problem under transportation cost uncertainty.**

In our setting, these disruptions are uncertain concerning their impact on transportation cost increases, duration, and time of occurrence during the planning horizon. Hence, the decision maker needs to anticipate the relevant transportation costs in delivery time periods in the future since the surcharge is not defined when the reorder is placed but when the replenishment is executed. This need to anticipate future cost developments depends on the specific supplier lead time offered. The longer the supplier lead time, the more time periods in the future costs need to be anticipated. No probability distribution for the future occurrence of disruptions is known; however, historical information is available.

Thus, a central decision maker manages inventory at the customer location and decides on the supplier to replenish from, how much to replenish, and when to replenish, given that a replenishment order can be made at each of the time periods. In the environment of direct deliveries, the replenishment quantity is a multiple of full-truck load equivalents. The overall objective is to minimize the total expected costs as sum of inventory holding, transportation, and shortage costs. Specifically, we minimize the trade-off between the costs of paying a cost premium to replenish for an alternative supplier or building inventory when no disruption occurs and the uncertain cost increase from a replenishment that is affected by a disruption.

# 4 Model development

In this section, we present a mathematical formulation for the SIRPDD. We first describe a deterministic model formulation in Section 4.1, and discuss the stochastic programming extensions for the transportation cost uncertainty in Section 4.2.

### 4.1 A deterministic formulation for the IRPDD

A set of suppliers $\mathcal{I}$ delivers a single product to the production facility over a discrete time horizon $\mathcal{T}$. Suppliers deliver in specific order sizes $o \in \mathcal{O}$ that define the replenishment quantities $q_{io}$, which are multiples of full-truck load equivalents. Each supplier uses a specific transportation mode that requires a delivery time $l_i$ at a transportation cost. These transportation costs $c^{tp}_{io(t+l_i)}$ occur upon arrival $(t + l_i)$, thus $l_i$ periods after a replenishment order was placed in order time period $t$ and include the supplier-specific product price. The customer holds inventory to satisfy the demands $d_t$. The overall inventory stored is limited by the inventory capacity $\hat{Y}$ with the starting inventory being $Y_0$. Besides transportation costs, the total costs consist of inventory holding costs $c^{ih}$ per product unit that is stored in each time period and a penalty costs of $c^{sh}$ per product unit that is not fulfilled.

The binary decision variable $x_{iot}$ decides in which time period $t$ to order from which supplier $i$ and in which order size $o$ to minimize the total costs. In addition, two auxiliary decision variables are taken into account that depend on the choice and timing of orders. First, $y_t$ is continuous and describes the inventory level at the customer location in time period $t$. Second, the continuous decision variable $p_t$ accounts for the production shortages due to missing raw materials in time period $t$ in case demands exceed the available inventory. Within this notation, the mixed-integer linear programming model for the deterministic problem is

$$\text{minimize} \quad \sum_{i\in\mathcal{I},o\in\mathcal{O},t\in\mathcal{T}} c^{tp}_{i(t+l_i)} \cdot q_{io} \cdot x_{iot} + \sum_{t\in\mathcal{T}} c^{sh} \cdot p_t + \sum_{t\in\mathcal{T}} c^{iv} \cdot y_t \tag{1a}$$

$$\text{subject to} \quad y_t \leq \hat{Y}, \qquad \forall t \in \mathcal{T} \tag{1b}$$

$$y_t = y_{(t-1)} + p_t - d_t + \sum_{i\in I} q_{io} \cdot x_{io(t-l_i)}, \quad \forall t \in \mathcal{T} \setminus \{0\} \tag{1c}$$

$$y_0 = Y_o, \tag{1d}$$

$$x_{iot} \in \{0,1\}, y_t \geq 0, p_t \geq 0, \qquad \forall i \in \mathcal{I}, o \in \mathcal{O}, t \in \mathcal{T} \tag{1e}$$

The objective function (1a) minimizes the total costs consisting of the transportation costs, the shortage costs as well as the inventory holding costs. Due to delivery time effects between the ordering from the customer and the delivery of the supplier, the time period of the relevant transportation costs are offset by the respective delivery time $(t + l_i)$. The inventory available in each time period is constrained to a maximum inventory capacity as outlined in constraints (1b) whereas the inventory balance is ensured in constraints (1c) and (1d). The replenishment quantity as defined by the ordering decision $x_{iot}$ and the order quantity $q_{io}$ that is arriving as inventory is offset by the delivery time $l_i$, thus highlighting that in time period $t$ replenishment orders placed in $t - l_i$ will arrive given that $l_i$ can differentiate by supplier $i$.

### 4.2 Multi-stage stochastic programming extension

We model the transportation cost uncertainty based on a scenario tree $\tau$ (cf. Ruszczyński and Shapiro, 2003; Huang and Ahmed, 2009) and assume knowledge of the probabilities of future cost developments at the beginning of each time period $t$. Using this modeling approach, we can capture the transportation cost uncertainty, that a decision maker faces at each order time period $t$, regarding the future transportation costs $c^{tp}_{io(t+l_i)}$, through scenarios. Thus, we consider a scenario tree, where a node $n$ in level $t$ of the tree corresponds to a specific realization of the uncertain parameter $c^{tp}_{io(t+l_i)n}$. Each level of the decision tree corresponds to a time period $t$, i.e., there are $|\mathcal{T}|$ levels in the tree. In addition, each node $n$ of the scenario tree, except the root node $(n = 1)$, has a unique parent $a(n)$ while each non-terminal node $n$ itself is the root of a subtree $\tau(n)$. The set $\varphi_t$ refers to all nodes corresponding to a time period $t$, while $t_n$ is the time period corresponding to node $n$. The probability of the realization of node $n$ is denoted by $\pi_n$ with $\sum_{n\in\varphi_t} \pi_n = 1$ whereas the sum of probability of all child nodes of a parent nodes equal the probability of the parent node, i.e., $\sum_{m\in\tau(n)} \delta_{a(m),n} \cdot \pi_m = \pi_n$ for all $n \in \tau$

with $\delta_{a(m),n}$ being the Kronecker delta. The path from the root node ($n = 1$) to a node $n$ is denoted by $\wp_n$. If $n$ is a terminal node ($n \in \varphi_T$), then $\wp_n$ corresponds to a *scenario*, and represents a realization of the transportation costs over all time periods $\mathcal{T}$. Let $\mathcal{S}$ be the number of corresponding scenarios (lead nodes) and $\mathcal{N}_\mathcal{T}$ the number of nodes in the whole tree. We summarized some of the notation associated with a scenario tree in Figure 2.

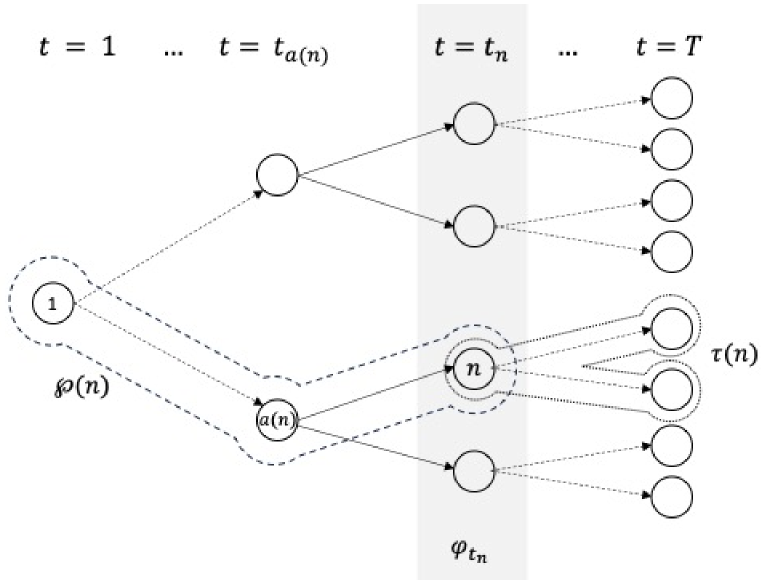


**Figure 2: Scenario tree notation, see Huang and Ahmed (2009).**

Assuming that the complete scenario tree describing all possible realizations and probabilities is available, we can formulate the multi-stage stochastic program as

$$\text{minimize} \quad z = \sum_{n \in \tau} \pi_n \cdot \Big( \sum_{i \in \mathcal{I}, o \in \mathcal{O}, t \in \mathcal{T}} c^{tp}_{i(t+l_i)n} \cdot q_{io} \cdot x_{iotn} + \sum_{t \in \mathcal{T}} c^{sh} \cdot p_{tn} + \sum_{t \in \mathcal{T}} c^{iv} \cdot y_{tn} \Big) \tag{2a}$$

$$\text{subject to} \quad y_{tn} \leq \hat{Y}, \qquad \forall t \in \mathcal{T}, n \in \mathcal{N} \tag{2b}$$

$$y_{tn} = y_{(t-1)a(n)} + p_{tn} - d_t + \sum_{i \in I} q_{io} \cdot x_{io(t-l_i)n}, \quad \forall t \in \mathcal{T} \setminus \{0\}, n \in \varphi_t \tag{2c}$$

$$y_{0n} = Y_0, \qquad \forall n \in \varphi_0 \tag{2d}$$

$$x_{iotn} \in \{0, 1\}, y_{tn} \geq 0, p_{tn} \geq 0 \qquad \forall i \in \mathcal{I}, o \in \mathcal{O}, t \in \mathcal{T}, n \in \mathcal{N} \tag{2e}$$

The above multi-stage model involves a replenishment decision $x_{iotn}$ with the resulting inventory levels $y_{tn}$ and product shortages $p_{tn}$ corresponding to each node $n$ of the scenario tree.

# 5 CFML framework

The goal of the replenishment problem is to determine the cost-optimal replenishment order decision, $a, f_{iot}(\mathcal{V}) \in \{0, 1\}$, for each supplier $i$ and each order size $o$ at each order period $t$ after having observed all features, $\mathcal{V}$. Unlike traditional learning frameworks, which maximize the ability to predict historical optimal replenishment decisions $a^*$, the CFML framework incorporates the objective function $z(a)$ of decision $a$ as defined in (2b) into the hyperparameter tuning of the learning framework itself. The core idea is that deviations from the binary labels should be evaluated in terms of their cost impact and not just a pure comparison so that, e.g., a replenishment order taken one day earlier or later might only have limited influence on total costs, and thus is less penalized in comparison.

The overall CFML framework is summarized in Figure 3. Let $\mathcal{K}$ be the index set of all training data points with $\mathcal{V}_k$ the corresponding feature vector. To obtain the mapping function $f(\cdot)$ that predicts the order decisions based on given features, we train the model with the labels. These labels are obtained from historical data and need to be generated first. The details of the so-called "labeling" step are described in Section 5.1. After labels are created, we train the supervised learning algorithm with the following training objective:

$$\min_f \sum_{k \in \mathcal{K}} z(f(\mathcal{V}_k)) \tag{3}$$

Thus, we do not aim to minimize the absolute difference $|f(V_k) - a_k^*|$ between the predicted and optimal binary decision. We describe the details on the mapping function $f$ as well as the hyperparameter tuning using a genetic algorithm in Section 5.1. Through the chosen learning function, the decision rules of the trained model can easily be extracted and visualized in the form of a replenishment policy (see Section 6.4.2).

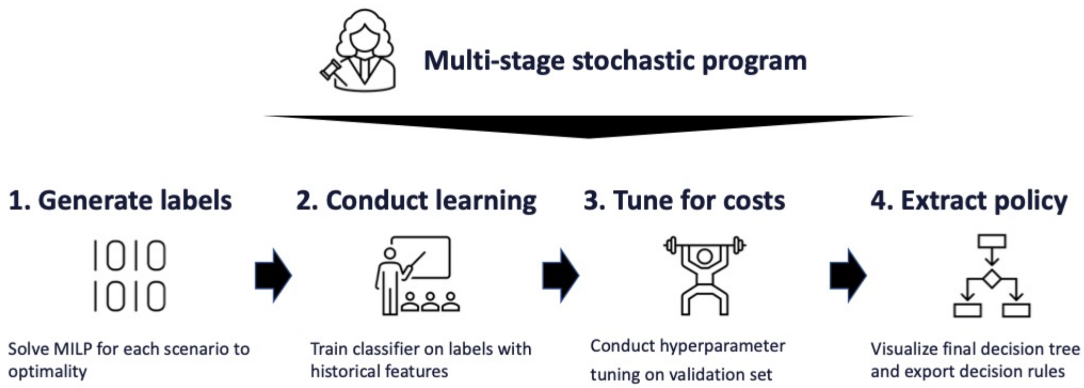


**Figure 3: Overview of CFML framework.**

## 5.1 Labeling the optimal replenishment policy

For a multiperiod inventory problem with multiple suppliers, lead time differences, and transportation cost uncertainty, the optimal replenishment policy cannot be defined in a straightforward way. Instead, we learn from historical data in a supervised way to derive such a replenishment policy that solves the SIRPDD. As a result, such a policy needs to define when to order from which supplier in which available order quantity.

In order to do so, we generate labels, i.e., order decisions, that over time represent an optimal replenishment policy. These labels cover various different disruption and decision situations so that a significant degree of the uncertainty space is covered. Thus, we derive scenarios $S$ with different planning horizons $\tilde{T}$ for the replenishment problem that describes this uncertainty. Still, it is sufficient to solve each scenario separately assuming perfect information and merge all generated labels afterwards for the training. This reduces the computational complexity and avoids the need of determining the full scenario tree. After solving each scenario, each replenishment order decision in each time period of the planning horizon serves as one label.

We obtain the different scenarios with limited historical data by adjusting both the starting inventory level $Y_0$ as well the planning horizon as a subset of the available historical data while remaining the time-series itself. As a result, scenarios include different decision situations so that even with perfect information, labels are generated for excess and shortage inventory situations. Thus, the different scenarios, must cover a variety of the situations that a decision maker would face in the uncertain future.

Beyond the labels, a supervised learning approach requires features to conduct the training upon. These features contain the available inputs at the time of decision, thus imperfect information. We refer to Section 6.2 for an overview of the available case-specific features that link to the replenishment reorder decision in each scenario $s$ and time period $t$.

## 5.2 CFML learning structure

After the associated labels for training data are obtained, we train a supervised learning model and tune its hyperparameters in a separate validation step. For training and tuning, the time-series of all historical labels is split at a certain time period into two time-series; a training set $T^1$ and a validation set $T^2$. This is done by choosing a time period $t^*$ with all observations in $T^1$ prior and all observations in $T^2$ after $t^*$. For each supplier $i$ and order size $o$, we train a classifier with a distinct hyperparameter set. Thus, the number of classifiers to train equals the sum of all order sizes $O$ across all suppliers $I$.

For each classifier, we use the Classification and Regression Tree (CART) (Breiman, 1984) to train the decision tree as this algorithm generally performs well in uni-variate split settings with small data sets (e.g. Lim et al., 2000; Patel and Prajapati, 2018). In the CART algorithm, at each node, rules that split the sample based on the feature values provided are chosen to minimize a pre-defined criterion. Several hyperparameters are studied that greatly influence the predictive performance of the CART algorithm (see Section 6.2). These hyperparameters need to be provided within the configuration of the CART algorithm. Hyperparameter tuning means identifying a set of hyperparameters that optimizes the performance of the CART algorithm with respect to a pre-defined criteria. In the case of our CFML framework, we choose hyperparameters that minimize the total costs of the resulting inventory replenishment policy (Equation (3)) on the validation set $T^2$.

In order to evaluate all possible combinations of values for the different hyperparameters, a large number of training and evaluation steps is required. Using the minimization of total costs as hyperparameter tuning objective, the total number of possible combinations rises dramatically compared to the hyperparameter tuning for traditional learning objectives as all hyperparameter combinations of all classifiers can only be evaluated together (see line 13 in Algorithm 1). Using the case and hyperparameter ranges as example, 1260 hyperparameter combinations need to be evaluated per classifier when conducting traditional hyperparameter tuning for predictive performance. As each classifier can be evaluated separetely, this results in about 3,000 combinations for three classifiers. In comparison, the hyperparameter tuning in the CFML results in $2 \cdot 10^9$ combinations as all combinations across the three classifiers need to be evaluated together (thus $1260^3$). In addition, the cost evaluation (see Equation (3)) is computationally more costly than comparing test labels against test predictions. In the case example, using full grid search results in intractable calculation time for hyperparameter tuning within the CFML due to the large state space.

To overcome this rise in computational complexity, genetic algorithms have already been successfully applied in hyperparameter tuning with the objective of minimizing an evaluation metric that quantifies the difference between predicted and test labels as part of a validation step (Alibrahim and Ludwig, 2021). Unlike these traditional approaches of hyperparameter tuning, we develop a genetic algorithm with the total costs (see Equation (2b) of the replenishment decision predictions as fitness function to find a well-performing hyperparameter set for the CFML. Thus, we do not aim to minimize the difference between predicted and test labels as part of hyperparameter tuning but the overall costs of applying the trained classifiers on a validation set. The pseudocode is outlined in Algorithm 1. We refer to Katoch et al. (2020) for a detailed overview on the genetic algorithm operators.

In a first step, we initialize a new population $P_1$ of size $n^P$ by random selection of hyperparameters for each classifier trained for each order size $o$ and supplier $i$. Thus, each individual in each population refers to a complete set of hyperparameters for all classifiers used. The main objective of this initialization step is to cover a diverse subset of all possible hyperparameter combinations. Then, for each new generation $g + 1$, the fittest individuals of previous populations are selected to inherit their genes

**Algorithm 1** **Genetic algorithm for hyperparameter tuning within CFML framework.**

**Input** generations $n^G$, population size $n^P$, mutation rate $\gamma$, sample size $\eta$, timeseries $T^1, T^2$

1: $P_1 \leftarrow$ Initialize population with $n^P$ random hyperparameter combinations<br>
2: **for** g=1 **to** $n^G$ **do**<br>
3:     $P_{g+1} = \{\}$<br>
4:     **while** $|P_{g+1}| \leq n^P$ **do**<br>
5:         **for** j = 1 **to** 2 **do** ▷ Generate parents<br>
6:             $z^* = \infty$<br>
7:             **for** n = 1 **to** $\eta$ **do**<br>
8:                 $p \leftarrow$ Select randomly from $P_g$<br>
9:                 **for each** $o \in \mathcal{O}$ and $i \in \mathcal{I}$ **do** ▷ Evaluate fitness of $p$<br>
10:                     $f_{io} \leftarrow$ Train with $T^1, p$ ▷ Train<br>
11:                     $\dot{x}_{iot} \leftarrow f_{io}(\mathcal{V}(t)) \; \forall t \in T^2$ ▷ Predict<br>
12:                 **end for**<br>
13:                 $z \leftarrow$ solve IRPDD **for** $x_{iot} = \dot{x}_{iot} \quad \forall t \in T^2, o \in O, i \in I$ ▷ Evaluate<br>
14:                 $p_j \leftarrow p$ **if** $z \leq z^*$<br>
15:             **end for**<br>
16:             $P_{g+1} \leftarrow P_{g+1} \cup \{p_j\}$<br>
17:         **end for**<br>
18:         child $\leftarrow$ Uniform Crossover($p_1$, $p_2$)<br>
19:         child $\leftarrow$ Mutate(child)<br>
20:         $P_{g+1} \leftarrow P_{g+1} \cup \{child\}$<br>
21:     **end while**<br>
22: **end for**

to the next generation with the objective of increasing the overall fitness across the population. To achieve this, a new and empty population $P_{g+1}$ is initialized in a first step. Then, while the number of individuals $|P_{g+1}|$ has not reached the population size $n^P$, two parents need to be generated in the second step. Each parent is determined out of a random sample of $\eta$ individuals drawn from the previous population $P_g$. We use the tournament selection in which only the fittest individual of the sample is chosen as parent (Miller and Goldberg, 1996). In order to assess the fitness of each individual $p$ (see lines $9-13$), we first train each machine learning classifier for each supplier $i$ and order size $o$ with the hyperparameters from $p$ on $T^1$ to obtain the mapping function $f_{io}$. Then, using this mapping function and the observations from $T^2$, order predictions $\dot{x}_{iot}$ are obtained. In the last step of the fitness evaluation, these order predictions for each order size $o$ and each supplier $i$ are evaluated together by solving the corresponding IRPDD on $T^2$ to obtain the total costs $z$. The best individual, i.e., with the lowest $z$, of the random sample is then a parent and saved to the next generation $P_{g+1}$. In addition, the characteristics of both parents $(p_1, p_2)$ are inherited to children through a cross-over. We use the uniform cross-over in which each parent has a random and equal probability of inheriting one of their genes, i.e., hyperparameter choices (Syswerda, 1989). Then, each new children is prone to a random mutation to any of the hyperparameters at a probability of $\gamma$. Once the number of individuals in $P_{g+1}$ reaches the population size $n^P$, a new generation is initialized and all steps repeat until the total number of generations $n^G$ is reached. The algorithm terminates by selecting the best-fit individual $p$ of the final population $P_{n^G}$ with the lowest cost $z$ on $T^2$.

# 6 Case study

We test our approach using a case study with real disruption data. Section 6.1 describes the case, while Section 6.2 outlines the numerical setup, labels, and case-specific features. In Section 6.3, we introduce replenishment policy benchmarks. Section 6.4 presents numerical results. Finally, in Section 6.5, managerial insights are drawn.

## 6.1 Case introduction

This case is based on a chemical company situated near the river Rhine, Germany. For the case company, more than 40% of all inbound materials are transported through the river Rhine, of which a large share is globally sourced. For these global shipments, only full-truck loads are considered due to the high share of transportation costs compared to inventory holding costs.

Two suppliers are available to deliver the product. The main supplier ($i = 1$) transports a regular order size ($o = 1$) that equals four days of average demand by vessel via the river Rhine and thus is prone to uncertain cost surcharges, which depend on the water level on the day of arrival. The alternative supplier ($i = 2$) uses a different transportation mode (i.e., rail) and is thus not subject to surcharges that depend on the water level. However, the overall transportation costs $c_{2t}^{tp}$ are 110% higher compared to the main supplier's disruption-free costs (thus $c_{2t}^{tp} = 2.1 \cdot c_{1t}^{tp}$). In addition, a larger order can be placed with the main supplier ($o = 2$), which equals 10 days of average demand to build up inventory and avoid potential surcharges. Thus, we differentiate between three types of replenishment orders in each time period $t$: The large order decision with the main supplier ($x_{11t}$), the regular order with the main supplier ($x_{12t}$), and the order with the alternative supplier ($x_{21t}$). Inventory storage capacity is limited to a storage capacity of five weeks of average demand. If the projected inventory exceeds the inventory capacity, the corresponding replenishment order is canceled.

Following industry standards, contractual agreements are in place between the shipment carriers that operate on the river Rhine and their customers that define the surcharges to be paid based on the water levels upon arrival. Due to the high lead times with the main supplier of $l_0 = 21$ days, the case company faces a high level of uncertainty concerning the surcharges to be paid based as they depend on the future water level in 21 days after ordering. To highlight the overall water level fluctuations, Figure 4 shows the water level history for the last eight years including the first threshold where surcharges occur in orange and where generally no transport is guaranteed anymore in red. For example, transportation stops occurred in 2015, 2017, 2018, 2020, and 2022, while high tides caused transportation stops in 2018 and 2021. In addition, occurrences where low water levels result in surcharges occur regularly. While some degree of seasonality seems visible, i.e., high tides happen exclusively in winter, while low water levels are typically seen in late summer or autumn, the overall fluctuations on a year-to-year level are significant.

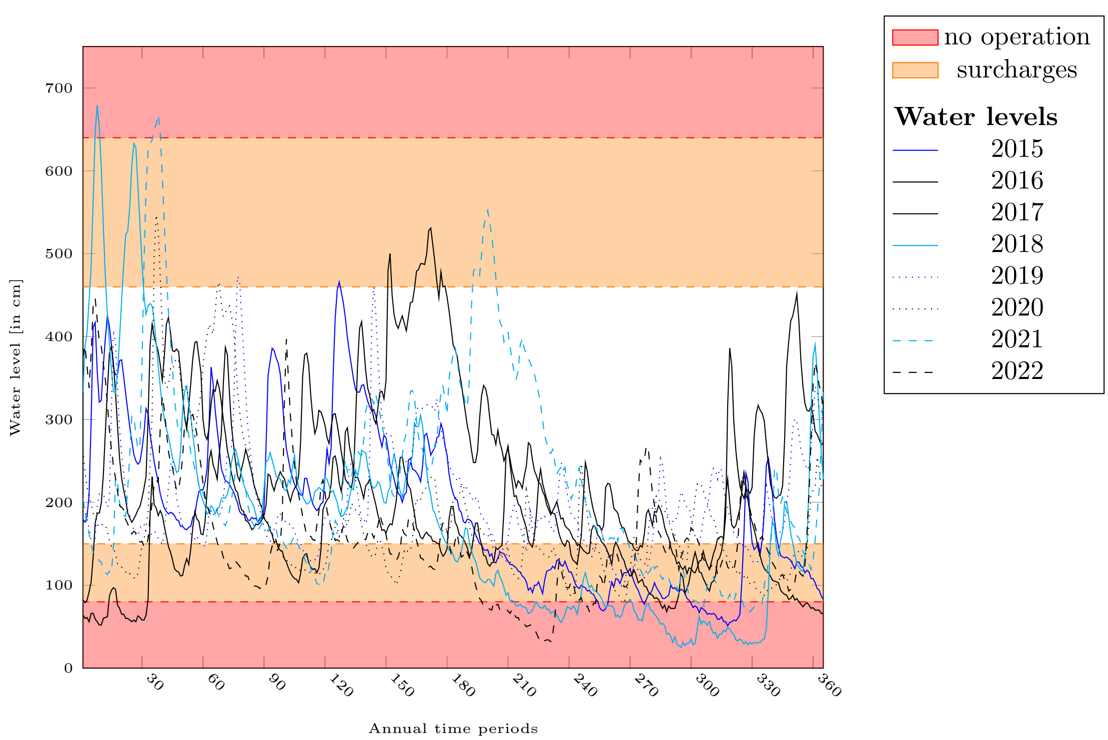


**Figure 4: Historical water levels at shipment critical point for river Rhine that define surcharges for main supplier ($i = 1$).**

If the water level falls below or is higher than the contractually defined thresholds, the resulting surcharges are significant. These surcharges range from 90 EUR to 300 EUR per TEU or even a shipment cancellation for water levels below 80cm and higher than 640cm. The significance of these surcharges also depends on the overall transportation costs of the container as well as the value of the goods transported when considering the trade-off between holding additional inventory at the benefit of avoiding these surcharges. The resulting total costs per TEU, including the surcharges between carriers and customers motivated by the case example, are summarized in Table 1.

**Table 1: Transportation costs per TEU transported including water level based surcharge.**

| Water level [cm] | $< 80$ | $[80, 90)$ | $[90, 100)$ | $[100, 110)$ | $[110, 130)$ | $[130, 150)$ | $[150, 460)$ | $[460, 640)$ | $\geq 640$ |
|---|---|---|---|---|---|---|---|---|---|
| Costs TEU [EUR] | - | 415 | 340 | 295 | 250 | 205 | 115 | 205 | - |

## 6.2 Numerical setup, features and labels

All numerical experiments were run on an Apple M1 Pro 8-Core processor with 16 GB of RAM. The algorithm is implemented in Python 3.8. For label generation, we solved the IRPDD using Gurobi 9.5 for each scenario to optimality within minutes. All machine learning models were implemented with Scikit-learn (Géron, 2022).

Historical data for all features, e.g., water levels, including trends, is available from May 2017 to September 2023. To evaluate our approach, we split available data into a training (2017-2019), a validation (2020-2021), and an evaluation set (2022-2023). In doing so, we simulate an application of the machine learning model in practice. Besides the water levels on the main transportation mode during the time period $t$, several additional features are needed and available. These are summarized in Table 2.

**Table 2: Labels and features used for machine learning approach.**

| Labels | |
|---|---|
| $x_{iot}$ | replenishment order of order size $o$ from supplier $i$ in time period $t$ |
| Features | |
| $y_t$ | inventory level at customer location in time period $t$ |
| $IP_t$ | inventory position at customer location in time period $t$ |
| $WL_t$ | water level at time period $t$ |
| $WL_t^{tr3}$ | historical trend of water level development between $t$ and $t-3$ |
| $WL_t^{tr7}$ | historical trend of water level development between $t$ and $t-7$ |
| $PWL_t^7$ | prediction of water level at $t+7$ |
| $PWL_t^{14}$ | prediction of water level at $t+14$ |
| $PWL_t^{tr7}$ | trend of change in prediction of water level at $t+7$ for the last three days |
| $PWL_t^{tr14}$ | trend of change in prediction of water level at $t+14$ for the last three days |
| $W_t$ | Calendar week of decision period |
| $M_t$ | Calendar month of decision period |

In a pre-processing step, we calculate the inventory position $IP_t$ based on the inventory level $y_t$, large orders placed with the main supplier ($x_{11t}$), regular orders placed with the main supplier ($x_{12t}$), orders placed with the alternative supplier ($x_{21t}$) as well as prospective shortages $p_t$. Using the inventory position instead of the inventory level as a reorder trigger is standard in most continuous-review stochastic inventory problems (Parlar, 1997). In addition to the current water level at the decision time period $t$, we calculate the last three and seven-day trends that indicate whether the water level has been rising or falling just before the decision time period $t$. Historical predictions of future water levels are available with a lag of seven ($PWL^{tr7}$) and 14 days ($PWL^{tr14}$). However, no predictions are available within the relevant lead time of 21 days. This lack of longer prediction horizons is typical as, generally, weather forecasts are available with a 14-day outlook. In addition, we create the trend of the water level predictions as a corresponding feature, thus the change of the 7-and

14-day forecast over the last three days ($PWL_t^{tr7}$ and $PWL_t^{tr14}$). Lastly, the calendar week $W_t$ and the calendar months $M_t$ of the decision time period is added.

Due to the label generation process of varying starting inventory levels and shifts in the planning horizon, training and testing data contain many dependent samples, such as multiple samples for the same date. Thus, even though starting inventories and horizons are varied, multiple labels with identical features (same week, months, water level, inventory position, etc.) exist. Given these multiple samples of identical labels as well as the connection between labels in a time-series, applying a random train test-split could result in dependent samples that exist in training and testing, and thus, result in overoptimistic prediction accuracies on testing as information is leaked. To handle the challenge of dependent samples, we build groups of labels through equal week-year combinations and ensure that each group is either present in training or testing but no label of a group in both. The year itself is dropped as a feature as it would not be helpful in making predictions for upcoming years. In addition, we use cross-validation to estimate the performance of the CFML on unseen data with limited samples. We conduct five folds of training and testing on the training data (years 2017-2019). We assess the costs for the fitness function as average across the trained predictors for each of the five folds when applied on the validation set (year 2020).

A further challenge is posed by the fact that the training data is highly imbalanced: The large main order labels contain 292 order and 19172 no order labels. Generally, this significant imbalance poses a challenge to any machine-learning approach. To overcome this, we conduct a random under-sampling of the majority class (Japkowicz, 2000). However, instead of forcing a fully balanced distribution of the classes, we conduct the under-sampling to match the disruption-free replenishment order cycle. For the case example, without disruption occurrence and due to the necessity of shipping full-truck loads, an order would be placed every five days. This balance is represented in each of the three different replenishment decisions.

The ranges for all relevant hyperparameters for tuning the decision tree predictors are summarized in Table 3. The criterion measures how the quality of a split in a decision tree is evaluated using either the gini criterion for homogeneity of the nodes or entropy for the information gain (James et al., 2021). The maximum depth of the tree sets the maximum number a of nodes a tree can have, i.e. how much the tree can expand, until all leaf nodes are pure or untill all leaf nodes contain less than the minimum number of samples per split. These minimum samples per split describes the threshold for an internal node to become a leaf node. Thus, a node can only become an internal node if the the number of samples are higher than the minimum samples per split. In addition, a split will only be considered if the resulting left and right branch of the split each contain at least the minimum number of samples per leaf. Finally, the maximum number of features to be considered each time to make a split decision can be limited up to the total number of features available. In case the maximum number of features is lower than the number of available features, a random selection of features is chosen at each split and evaluated concerning the best feature to use for the split. The number of features in the selection matches the max feature. This mechanism is often used to control overfitting.

**Table 3: Decision tree classifier hyperparameter ranges for tuning.**

| Hyperparameter | Possible values |
|---|---|
| Criterion | Gini, Entropy |
| Maximum depth | None, 3, 4, 5, 8, 12 |
| Min samples split | 2, 6, 10, 20, 50 |
| Min samples leaf | 1, 4, 6, 10, 15, 20, 30,40 |
| Max features | 2, 4, 6, 11 |

Within the CFML, we run the genetic algorithm with $\delta = 10$ generations, a population size of $\alpha = 100$, a mutation rate of $\gamma = 0.1$, two parents $\beta = 2$, and a random sample size of $\eta = 6$. The genetic algorithm terminates within less than three hours.

## 6.3 Comparison with standard policies and traditional machine learning frameworks

To demonstrate the performance of our CFML framework, we compare the obtained replenishment policy against inventory replenishment policies obtained from the literature and alternative machine learning frameworks. Generally, the inventory replenishment problem under disruption uncertainty with multiple suppliers that differ in their lead time would require a multiple dimensional state space (Chen et al., 2012). Thus, literature dealing with stochastic transportation or purchasing costs is sparse (Darwish, 2008). As a result, we consider a general purpose replenishment policy.

The *reorder point - reorder quantity* policy, often referred to as (s,Q) is widely used in practice and extensively studied in literature. They are standard in inventory systems with uncertain demand and delivery times. However, as demands are stable and delivery times are known, the policy can be simplified to triggering an order whenever the inventory level reaches the reorder point that marks the demand during the delivery time of the supplier.

The *risk averse (RA)* policy only orders with the reliable alternative supplier and thus represents the risk-averse idea of not placing orders with the main supplier as there might be the risk of paying surcharges upon arrival of the orders. As a result, transportation costs are not prone to disruptions and are known upfront. However, a cost premium is paid compared to the surcharge-free costs of ordering with the main supplier.

In addition, we benchmark our CFML framework against machine learning frameworks that maximize each classifier's individual ability to predict the training set's order decisions. Notably, we consider a neural network (ML-NN), a logistic regression (ML-LR), and a decision tree (ML-DT). All machine learning models are tuned using the f1-score as the objective for selecting the best-fit hyperparameters during cross-validation. The f1-score is the harmonic mean of the precision (i.e. positive predictive value) and recall (i.e. sensitivity) with 1.0 being the highest possible value. Even though we use undersampling (see Section 6.2), the positive class still contains fewer samples than the negative class. The f1-score can help to balance the metric across positive and negative samples and thus, is a better fit than a simple accuracy metric that would already achieve a high accuracy by solely predicting the negative class. Besides this diverging learning objective in hyperparameter tuning, they are trained with the exact same undersampled labels as the CFML as well as the same grouping in k-fold cross validation.

## 6.4 Results

In this section, we apply the decision rules obtained through the CFML and other benchmark policies to the unseen evaluation set to compare the cost performance and analyze differences in the order patterns. Then, to understand the cost differences between the CFML and the ML-DT that use identical hyperparameter ranges and learning algorithms, we compare the prediction performances on the test set and analyze how optimal hyperparameter settings differentiate. In addition, we visualize the results of the tuned ML-DT classifier and discuss challenges of traditional learning frameworks for this problem setting.

### 6.4.1 Cost performance out-of-sample

All replenishment policies are benchmarked against the *perfect information scenario (PI)* that assumes full knowledge of the future transportation cost development. Even though unrealistic as it ignores the problem characteristic of uncertain transportation costs in the future, it gives the lowest bound on the costs of the SIRPDD. Table 4 summarizes the results on the out-of-sample evaluation set. This evaluation set contains surcharges on the main transportation mode based on the water levels from 2022-2023 while all machine learning models are trained using labels from 2017-2021.

Table 4: Comparison of different replenishment policies on the total planning horizon.

| | | PI | CFML | ML-DT | ML-NN | ML-LR | (s,Q) | RA |
|---|---|---|---|---|---|---|---|---|
| Obj | Costs | 323.79 | 466.43 | 569.96 | 764.11 | 634.75 | 582.18 | 599.58 |
| | Increase | 0% | 44% | 76% | 136% | 96% | 80% | 85% |
| Orders | Large main | 14 | 20 | 39 | 0 | 35 | 0 | 0 |
| | Normal main | 97 | 90 | 48 | 153 | 60 | 147 | 0 |
| | Alternative | 12 | 5 | 0 | 0 | 6 | 0 | 147 |

Starting with the CFML, our new learning framework significantly outperforms all other policies. It represents a cost reduction of roughly 20% compared to the (s,Q) policy, which reflects the current industry standard, and 18% compared to classical machine learning frameworks. Compared to the PI solution, the CFML represents a cost increase of 44%. Though this indicates further improvement potential, it is in line with recent work of Qi et al. (2023) who determined optimal replenishment quantities for a single supplier in fixed replenishment cycles, thus a more narrow problem setting, and report a gap of 25% to their perfect information setting.

In addition, the split of the order predictions of the CFML approximates the ideal split of the perfect information (PI) solution though differences remain particularly in the number of alternative order decisions. Notably, the split of the order predictions of the CFML approximates the ideal split of the perfect information (PI) solution. Still, differences in the pure quantities can be observed, while the exact timing of replenishment orders explains the remaining gap. This remaining 44% gap to the perfect information solution highlights the high problem complexity and degree of transportation cost uncertainty that is characteristic for the problem setting of determining the optimal supplier, at the optimal quantity, in the optimal time.

Compared to the traditional machine learning frameworks that are tuned for prediction performance using a neural network (ML-NN) and a logistic regression (ML-LR), we can see that both frameworks result in replenishment order predictions that perform significantly worse than the RA or (s,Q) policy and thus, do not seem a good fit for this specific problem setting. Specifically, the neural network seems to suffer from a severe overfitting reaction as nearly whenever inventory capacity allows, a regular main order is triggered (153 main regular orders). While the logistic regression performs slightly better regarding total costs (634.75), it is also not a good fit for this specific problem setting. In comparison, the ML-DT and the CFML that use a decision tree as a classifier perform better than the remaining benchmarks. The ML-DT, which uses a decision tree, performs slightly better than the fixed replenishment rules (s,Q) and RA; however, it represents a 22% cost increase compared to the CFML and highlights the need for the revised learning objective highlighted in Equation 3. In the ML-DT, no alternative order is released during the evaluation period, and compared to the PI solution, significantly more large main orders are triggered.

Lastly, we compare all results against two fixed replenishment rules. The risk-averse policy of ordering only with the alternative supplier leads to the highest total costs and a cost increase of 85% compared to the replenishment with perfect information. Instead, triggering regular order sizes with the main supplier through an (s,Q) policy reduces the costs by 3% even though significant surcharges need to be paid. Without surcharges, the alternative supplier marks a cost increase of 110% compared to the main supplier. However, even though significant surcharges are paid, the (s,Q) policy still performs slightly better than the pure RA policy. Overall, both fixed replenishment rules do not seem a good fit for this problem setting and highlight the potential of extracting problem-specific rules using the CFML learning framework.

### 6.4.2 Prediction performance

To recap, the CFML and the ML-DT are trained on the identical and undersampled data set using group k-fold cross-validation. Still, as seen in Section 6.4.1, the CFML significantly outperforms the

ML-DT. Thus, the key differentiation is the diverging hyperparameter tuning objective, which is the f1-score (average across all folds) for ML-DT and the resulting policy's effectiveness on the validation set for CFML. In the first step, we compare the average prediction performance across the different folds for all three decisions separately. Key prediction scores are summarized in Table 5, and the confusion matrix results for a testing fold are shown in Table 6.

**Table 5: Performance scores on testing split.**

| | Accuracy | | f1-score | |
|---|---|---|---|---|
| | CFML | ML-DT | CFML | ML-DT |
| Large main order | 69.52% | 62.58% | 2.73% | 57.01% |
| Regular main order | 96.55% | 96.56% | 92.79% | 92.98% |
| Alternative order | 92.24% | 99.79% | 88.71% | 99.26% |

As one could expect given that the ML-DT is tuned for f1-score maximization, the ML-DT outperforms the CFML regarding the observed f1-score across all three decision classifiers. However, three interesting observations can be made. First, even though the f1-score is significantly higher for the large main order decision, the CFML outperforms the ML-DT in terms of prediction accuracy. This can be explained as due to the undersampling three times more negative than positive labels remain in the dataset. Thus, as can be seen in greater detail in the confusion matrix in Table 6, the ML-DT aims at a balanced prediction of positive and negative labels while the CFML seems to favor an accurate prediction of negative labels. Second, there is no significant difference in prediction scores for the regular order decision with the main supplier between the CFML and the ML-DT. This observation fits the characteristic of the main regular decision as it represents the replenishment of a decision maker without the transportation cost uncertainty. As a result, both tuning approaches result in similar predictions. Lastly, the trained classifier for the alternative order outperforms the CFML classifier in terms of f1-score and prediction accuracy. Overall, it is possible to accurately predict optimal regular main order and alternative order decisions using decision trees, while the large main order decision seems inherently more challenging to predict. In addition, based on the prediction scores, one would expect the ML-DT to outperform the CFML as the higher prediction scores should be reflected in an overall cost-optimal replenishment policy.

Comparing the confusion matrix results (see Table 6), it is clear that the lower prediction scores for the large main order decision are due to the tendency of the CFML classifier for a negative decision. On the testing split, less than 15% of the predictions are positive, while interestingly, the opposite can be seen with the ML-DT. Here, all positive labels in the testing split are predicted accurately and another 87 labels are predicted as false-positive. Likely, this results in predictors that favor the alternative order decision over the inventory build-up decision for the CFML. In contrast, the opposite seems to hold true for the ML-DT. Overall, the hyperparameter sets themselves seem to play a major role in the classifier performance.

**Table 6: Confusion matrix on testing split.**

| | Confusion Matrix - CFML | | | | | | Confusion Matrix - ML-DT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Large main | | Regular main | | Alternative | | Large main | | Regular main | | Alternative | |
| Pred. | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 151 | 28 | 2044 | 20 | 303 | 53 | 95 | 87 | 2028 | 26 | 349 | 2 |
| 1 | 48 | 6 | 71 | 638 | 0 | 123 | 0 | 52 | 52 | 646 | 0 | 129 |

Thus, besides the diverging predictive performance, it is interesting to understand how the optimal hyperparameter sets differentiate for the three decisions. The corresponding optimal hyperparameters for the CFML and the ML-DT are shown in Table 7. Following the overall similar prediction score performance for the regular main order, the optimal hyperparameter set for the regular main order decision also nearly matches between the CFML and the ML-DT. Specifically, the main depth, maximum number of features to be considered per split, and the minimum samples per split match exactly.

The key difference can be observed in the large main order decision. For this decision, the CFML selects each hyperparameter differently than the ML-DT. Important to note is that the CFML favors a combination of a tree that is not limited in depth (max depth=none) and considers many features at each split while only considering splits that contain a high number of 50 minimum samples. The hyperparameter set of the ML-DT, on the other end, highlights the difficulty of accurately balancing the positive and negative predictions as the optimal set only considers two random features at each split (max features 2) and is of limited depth (max depth = 5).

**Table 7: Results hyperparameter tuning.**

| Order | Model | Criterion | Max depth | Min samples split | Min samples leaf | Max features |
|---|---|---|---|---|---|---|
| Large main | CFML | Gini | None | 50 | 6 | 6 |
| | ML-DT | Entropy | 5 | 30 | 10 | 2 |
| Regular main | CFML | Entropy | 3 | 50 | 1 | 6 |
| | ML-DT | Entropy | 3 | 40 | 20 | 6 |
| Alternative | CFML | Entropy | None | 50 | 40 | 11 |
| | ML-DT | Entropy | None | 20 | 6 | 11 |

The fundamental difference between the CFML and the ML-DT seems to be in the different hyperparameters and thus decision trees obtained for the large main order decision. One of the key advantages of decision trees for practical application is that the outcome can relatively easily be visualized. We will use this advantage to understand the resulting tree of the ML-DT for the large main order decision in greater detail. The final decision tree for the large main order decision of the ML-DT approach is visualized in Figure 5. Positive predictions (class = y[1]) are highlighted in blue while negative predictions are highlighted in orange. Each split to the left equals a true statement, i.e., that the condition is fulfilled, while the opposite statement is false. Thus, each positive or negative prediction can be obtained by combining each condition. For example, looking at the bottom left positive prediction, the decision maker should place a large main order if the calendar week is higher than 12, the current inventory position is higher than 75, the current water level is below or equal to 306cm, it is earlier than June (calendar months $\leq$ 5.5), and the current calendar week is 13 or less. To summarize, we would place an order if we are in calendar week 13, the inventory position is higher than 75 units, and the current water level is below 306cm. While this represents a decision rule that performs well in cross-validation, it would be counter-intuitive for a human decision maker. Suppose the observed current water level is below the surcharge threshold and the 14-day prediction even forecasts lower water levels. This situation indicates a high likelihood of low water levels for the next 21 days, thus a high chance of surcharges to be paid with the main supplier. Still, the final policy obtained by the decision tree would result in an order placement with the objective of increasing inventory, which might not be cost-optimal. However, this ability to sense-check and even adapt such a rule in practice represents one of the significant advantages of decision trees for their application in practice. Interestingly, the ML-DT decision tree considers neither the forecast nor a water level trend feature.

## 6.5 Managerial insights

To summarize, the following managerial insights are drawn: (1) Through a combination of mathematical optimization with machine learning, replenishment policies for the multi-supplier inventory replenishment problem with transportation cost uncertainty and lead time differences can be obtained that outperform benchmark policies. This data-driven approach requires no knowledge of the underlying stochastic uncertainty but only historical disruption data. Publicly available databases, such as historical water level forecasts, can easily be incorporated as features by linking them to the respective labels to evaluate their value and contribution to future replenishment decisions within the supervised learning procedure. (2) Due to the complexity and inter-dependency of decisions in a multi-supplier replenishment problem with delivery time differences, multiple replenishment quantities, and transportation cost uncertainty, the traditional hyperparameter tuning approach for supervised learning

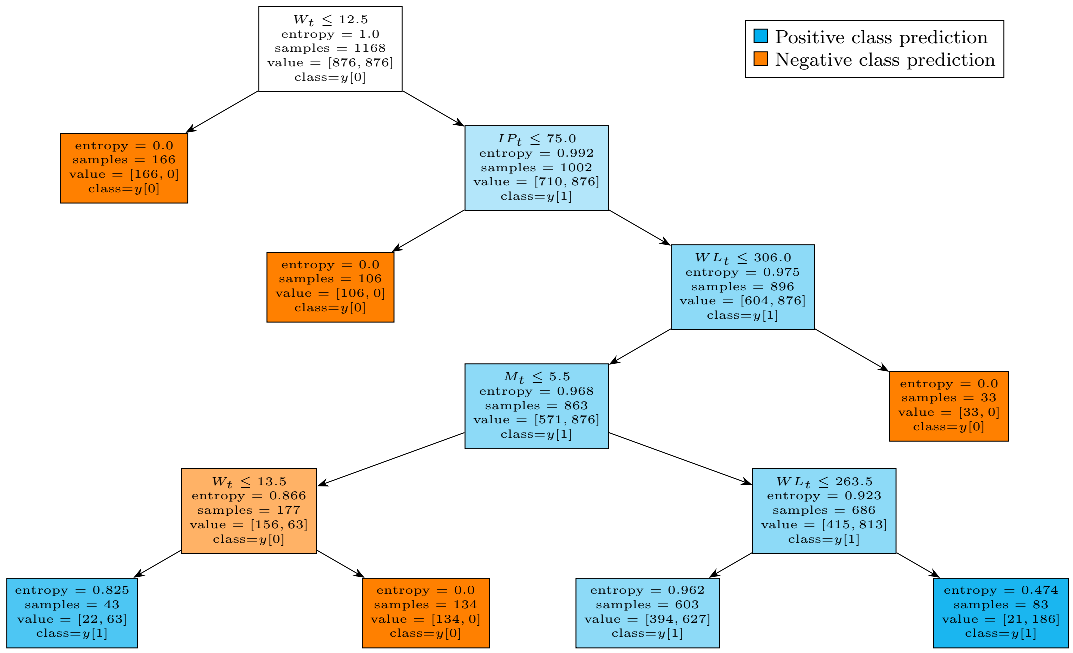


**Figure 5: Resulting decision tree large main order tuned for prediction accuracy.**

that maximizes individual decision prediction performance leads to improper replenishment policies. In the case example, the resulting policy even results in a cost increase compared to the risk-averse decision of only ordering from the resilient alternative supplier. Thus, problem-specific approaches, such as the CFML, must be explored further. (3) In this problem setting, the best-performing machine learning framework "only" uses decision trees as supervised machine learning model, which is easy to visualize and interpret in its output. When comparing the ML-DT with ML-NN, the neural network performs significantly worse than the decision tree, indicating that a more complex machine learning model is not helpful for this problem setting. Instead, a well-performing classifier can be trained using decision trees, allowing a human planner to directly understand the proposed decision rules and interact with the replenishment proposals as they can be visualized in a tree structure. As highlighted in Figure 5, this also helps to identify results that suffer from overfitting and allows an expert evaluation of the trained classifiers. Thus, potential data issues and flaws can be identified before the CFML application in practice. Not only are the rules transparent to planners, but if needed, they can be easily adjusted by manipulating the values of the if-else splits.

# 7 Conclusion and outlook

We have introduced a new problem setting for the inventory replenishment problem with multiple suppliers, lead time differences, and transportation cost uncertainty to decide when to source, which quantity, from which supplier. To obtain the required decision rules, we have developed a new machine learning framework, CFML, that directly optimizes the total costs of the different replenishment decisions within the hyperparameter tuning. We have tested the CFML in a case using real-life disruption data and shown that it significantly outperforms standard replenishment policies and traditional machine learning frameworks for inventory replenishment. Particularly, we have shown that the traditional hyperparameter tuning approach of optimizing prediction performance for each replenishment decision leads to inefficient policies. Furthermore, due to the use of decision trees as the supervised machine learning approach, the resulting policy is transparent and, if needed, can easily be adjusted by subject matter experts. Finally, we have highlighted managerial implications on the data-driven inventory replenishment with multiple suppliers and cost uncertainty, leveraging public databases for waterway transport disruption uncertainty.

Our study highlighted the enormous potential for data-driven solutions in the complex field of inventory replenishment with multiple suppliers and supply uncertainty; however, it is not without limitations. We have highlighted the benefits of our CFML for a single product and presented a solution procedure that develops these rules within a couple of hours on a personal computer. As each decision tree may depend on product characteristics, deploying the proposed approach at scale might include the necessity of handling thousands of stock-keeping-units and thus become computationally challenging. Thus, applying a clustering logic based on product characteristics to determine replenishment policies per cluster will be an exciting research area to allow an application in practice. Another interesting research area would be the planner acceptance and collaboration between planners and experts with a replenishment policy obtained in a purely data-driven way.

# A Appendix

As highlighted in Section 6.3, we compare the performance of our CFML using a decision tree as classifier against traditional learning approaches using a decision tree, a neural network, and a logistic regression. Compared to the CFML, each machine learning model is tuned with the objective of maximizing the f1-score during the hyperparameter tuning using a full grid search. In contrast to the CFML, the full grid search takes less than an hour for each classifier trained.

Regarding the hyperparameter ranges, the ML-DT is trained using the same hyperparameter ranges as the CFML. The respective parameters for the ML-NN are summarized in Table 8. For the logistic regression, the default hyperparameters of the Scikit-learn environment are used.

**Table 8: Neural network hyperparameter ranges for tuning.**

| Hyperparameter | Possible values |
|---|---|
| Hidden layer sizes | (10,30,10),(20,), (50,50,50), (50,100,50), (100,) |
| Activation | tanh, relu |
| Solver | sgd, adam |
| Alpha | 0.0001, 0.05 |
| Learning rate | constant, adaptive |

# References


Alibrahim, H. and Ludwig, S. A. (2021). Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization. In 2021 IEEE Congress on Evolutionary Computation (CEC).

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2021). Inventory routing under stochastic supply and demand. Omega, 102:102304.

Baryannis, G., Dani, S., and Antoniou, G. (2019a). Predicting supply chain risks using machine learning: The trade-off between performance and interpretability. Future Generation Computer Systems, 101:993–1004.

Baryannis, G., Validi, S., Dani, S., and Antoniou, G. (2019b). Supply chain risk management and artificial intelligence: state of the art and future research directions. International Journal of Production Research, 57(7):2179–2202.

Bassiouni, M. M., Chakrabortty, R. K., Hussain, O. K., and Rahman, H. F. (2023). Advanced deep learning approaches to predict supply chain risks under COVID-19 restrictions. Expert Systems with Applications, 211:118604.

Bertsimas, D., Shtern, S., and Sturt, B. (2023). A data-driven approach to multistage stochastic linear optimization. Management Science, 69(1):51–74.

Bollapragada, R. and Rao, U. S. (2006). Replenishment planning in discrete-time, capacitated, non-stationary, stochastic inventory systems. IIE Transactions, 38(7):605–617.

Breiman, L. (1984). Classification And Regression Trees. Routledge, 1 edition.

Brintrup, A., Pak, J., Ratiney, D., Pearce, T., Wichmann, P., Woodall, P., and McFarlane, D. (2020). Supply chain data analytics for predicting supplier disruptions: a case study in complex asset manufacturing. International Journal of Production Research, 58(11):3330–3341.

Chen, J., Zhao, X., and Zhou, Y. (2012). A periodic-review inventory system with a capacitated backup supplier for mitigating supply disruptions. European Journal of Operational Research, 219(2):312–323.

Chen, X., Deng, T., Shen, Z.-J. M., and Yu, Y. (2023). Mind the gap between research and practice in operations management. IISE Transactions, 55(1):32–42.

Chopra, S., Reinhardt, G., and Dada, M. (2004). The effect of lead time uncertainty on safety stocks. Decision Sciences, 35(1):1–24.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014). Thirty years of inventory routing. Transportation Science, 48(1):1–19.

Darwish, M. (2008). Joint determination of order quantity and reorder point of continuous review model under quantity and freight rate discounts. Computers & Operations Research, 35(12):3902–3917.

Gallego, G. and Simchi-Levi, D. (1990). On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer R-systems. Management Science, 36(2):240–243.

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.

Güllü, R., Önol, E., and Erkip, N. (1999). Analysis of an inventory system under supply uncertainty. International Journal of Production Economics, 59(1):377–385.

Hosseini, S., Ivanov, D., and Dolgui, A. (2019). Review of quantitative methods for supply chain resilience analysis. Transportation Research Part E: Logistics and Transportation Review, 125:285–307.

Huang, K. and Ahmed, S. (2009). The value of multistage stochastic programming in capacity planning under uncertainty. Operations Research, 57(4):893–904.

Iakovou, E., Vlachos, D., and Xanthopoulos, A. (2010). A stochastic inventory management model for a dual sourcing supply chain with disruptions. International Journal of Systems Science, 41(3):315–324.

Ivanov, D., Dolgui, A., and Sokolov, B. (2019). The impact of digital technology and industry 4.0 on the ripple effect and supply chain risk analytics. International Journal of Production Research, 57(3):829–846.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). An Introduction to Statistical Learning: with Applications in R. Springer US.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In Int'l Conf. on Artificial Intelligence, pages 111–117. Citeseer.

Kalymon, B. A. (1971). Stochastic prices in a single-item inventory purchasing model. Operations Research, 19(6):1434–1458.

Karlin, S. (1960). Dynamic inventory policy with varying stochastic demands. Management Science, 6(3):231–258.

Katoch, S., Chauhan, S. S., and Kumar, V. (2020). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80(5):8091–8126.

Koetse, M. J. and Rietveld, P. (2009). The impact of climate change and weather on transport: An overview of empirical findings. Transportation Research Part D: Transport and Environment, 14(3):205–221.

Kosasih, E. E. and Brintrup, A. (2021). A machine learning approach for predicting hidden links in supply chain with graph neural networks. International Journal of Production Research, 60(17):5380–5393.

Kouvelis, P. and Li, J. (2008). Flexible backup supply and the management of lead-time uncertainty. Production and Operations Management, 17(2):184–199.

Lim, T.-S., Loh, W.-Y., and Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning, 40(3):203–228.

Miller, B. L. and Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. Evolutionary Computation, 4(2):113–131.

Mišić, V. V. and Perakis, G. (2020). Data analytics in operations management: A review. Manufacturing & Service Operations Management, 22(1):158–169.

Neghab, D. P., Khayyati, S., and Karaesmen, F. (2022). An integrated data-driven method using deep learning for a newsvendor problem with unobservable features. European Journal of Operational Research, 302(2):482–496.

Nikolopoulos, K., Punia, S., Schäfers, A., Tsinopoulos, C., and Vasilakis, C. (2021). Forecasting and planning during a pandemic: COVID-19 growth rates, supply chain disruptions, and governmental decisions. European Journal of Operational Research, 290(1):99–115.

Parlar, M. (1997). Continuous-review inventory problem with random supply interruptions. European Journal of Operational Research, 99(2):366–385.

Patel, H. H. and Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering, 6(10):74–78.

Pettit, T., Fiksel, J., and Croxton, K. (2010). Ensuring supply chain resilience: development of a conceptual framework. Journal of business logistics, 31(1):1–21.

Priore, P., Ponte, B., Rosillo, R., and de la Fuente, D. (2019). Applying machine learning to the dynamic selection of replenishment policies in fast-changing supply chain environments. International Journal of Production Research, 57(11):3663–3677.

Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., and Shen, Z.-J. M. (2023). A practical end-to-end inventory management model with deep learning. Management Science, 69(2):759–773.

Ruszczyński, A. and Shapiro, A. (2003). Stochastic programming models. In Handbooks in Operations Research and Management Science, pages 1–64. Elsevier.

Saghafian, S. and Oyen, M. V. (2012). The value of flexible backup suppliers and disruption risk information: Newsvendor analysis with recourse. IIE Transactions, 44(10):834–867.

Schmitt, T. G. (1984). Resolving uncertainty in manufacturing systems. Journal of Operations Management, 4(4):331–345.

Silver, E. A. (1981). Operations research in inventory management: A review and critique. Operations Research, 29(4):628–645.

Svoboda, J., Minner, S., and Yao, M. (2021). Typology and literature review on multiple supplier inventory control models. European Journal of Operational Research, 293(1):1–23.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In Proceedings of the Third International Conference on GeneticAlgorithms, pages 2–9, San Matea, Ca. Morgan Kaufmann.

Whybark, D. C. and Williams, J. G. (1976). Material requirements planning under uncertainty. Decision Sciences, 7(4):595–606.

Williams, B. D. and Tokar, T. (2008). A review of inventory management research in major logistics journals. The International Journal of Logistics Management, 19(2):212–232.

Xiao, G., Yang, N., and Zhang, R. (2015). Dynamic pricing and inventory management under fluctuating procurement costs. Manufacturing and Service Operations Management, 17(3):321–334.

Xiong, X., Li, Y., Yang, W., and Shen, H. (2022). Data-driven robust dual-sourcing inventory management under purchase price and demand uncertainties. Transportation Research Part E: Logistics and Transportation Review, 160:102671.