Towards Supervised Learning of Optimal Replenishment Policies

Alexander Beier,[1] **Moritz Fleischmann**[2] **Heiner Stuckenschmidt,**[3]
[1, 3]Data and Web Science Group, University of Mannheim, B6 26, 68159 Mannheim, Germany
[2]Business School, University of Mannheim, Schloss, 68131 Mannheim, Germany
[1]rbeier@mail.uni-mannheim.de, [2]mfleis@mail.uni-mannheim.de, [3]heiner@informatik.uni-mannheim.de

Abstract

We propose a supervised learning algorithm for the multi-period inventory problem (MPIP) that tackles shortcomings of existing multi-step, model-based methods on the one and policy-free reinforcement learning algorithms on the other hand. As a model-free end-to-end (E2E) method that takes advantage of auxiliary data, it avoids pitfalls like model misspecification, multi-step error accumulation and computational complexity induced by a repeated optimization step. Furthermore, it manages to leverage domain knowledge about the optimal solution structure. To the best of our knowledge, this is one of the first supervised learning approaches to solve the MPIP and the first one to learn policy parameters. Given the variety of settings in which OR researchers have developed well-performing policies, our approach can serve as a blueprint of how to design E2E methods that leverage that knowledge. We validate our hypotheses on synthetic data and demonstrate the effect of individual model characteristics.

Motivation & Related Work

As the field of supply chain management as a whole, the subtask of optimizing a firm's inventory levels and order quantities in the presence of uncertain demand has been a central subject in operations research for decades, and its importance has developed from a "necessary evil" to a source of competitive advantage (Snyder and Shen, 2019, Ch. 1). First seminal works on the multi-period inventory problem (MPIP) date back as far as Arrow, Harris, and Marschak (1951), and Scarf (1960). The latter was the first to prove the optimality of an (s, S) replenishment policy. Under this policy, an order is placed to raise inventory to a target level S whenever the inventory position drops below a reorder point s. The author showed that the optimal, period-dependent policy parameters (s_t, S_t) can be computed by solving a dynamic program, and that the optimality result extends to the case of a positive, deterministic lead time. Kaplan (1970) and Ehrhardt (1984) generalized the (s, S)-optimality to stochastic lead times in the finite an infinite horizon setting, respectively.

More recently, OR researchers have been focusing on data-driven algorithms for inventory management, with an increasing emphasis on supervised learning approaches (Qi, Mak, and Shen 2020). Those efforts have led to convincing results especially in single-period problems, such as the Newsvendor model. Both shallow and deep neural networks have proven effective in predicting optimal order quantities from auxiliary data (Huber et al. 2019; Oroojlooyjadid, Snyder, and Takác 2016). Other successful machine learning (ML) methods include clustering approaches like k-nearest-neighbor (kNN), classification and regression trees (CART), random forests (Bertsimas and McCord 2019), and modified kernel regression methods (Ban and Rudin 2018). All of these methods exploit some notion of geometrical proximity between feature vectors to approximate the demand distribution non-parametrically in a first step, and then arrive at an ordering decision by completing the optimization step.

The common denominator of all data-driven approaches and their source of power is the effective use of meaningful feature data and the renunciation of model-based decision-making, which is known to invite biases from model misspecification (Huber et al. 2019). The E2E methods combine this advantage with the integration of the prediction and the optimization step, thereby avoiding an accumulation of inaccuracies common to multi-step decision-making (Qi, Mak, and Shen 2020). The success of those approaches raises the question whether similar ideas may perform well for multi-period problems.

However, transferring the available methods to multi-period problems, such as the MPIP, is not straightforward. In essence, the challenges can be broken down into two root problems. First of all, the single-period predict-then-optimize (PTO, (Qi et al. 2020)) methods take advantage of the fact that the solution of the Newsvendor Problem (the famous Newsvendor quantile) is well-known. Depending on the specific setting, however, the optimal solution for the multi-period counterpart can be mathematically intractable (Qi et al. 2020). Second, even in settings in which an optimal solution can be computed, other intricacies inherent to the MPIP pose problems. A direct transfer of the single-period PTO approaches suffers from computational complexity and would require knowledge of auxiliary data for the entire optimization horizon ahead of time. While the single-period E2E models do not necessarily require an explicit optimal solution, it is not obvious how to generalize them as they now need to be aware of the problem-inherent intertemporal dependencies. As a consequence, the field is dominated by two kinds of approaches. First, there are methods – mainly from the realm of reinforcement learning – that discard any potential prior knowledge about well-performing policies, may they be optimal or heuristic, and search for a close-to-optimal policy themselves. The obvious disadvantage is the disregard of useful structural information about the solution. Consequently, and despite considerable computational effort, RL policies may be outperformed by simplistic heuristics (Gijsbrechts et al. 2019). The second group consists of model-based, multi-step methods that solve the dynamic optimization program, either approximately or optimally, or implement a well-performing parametrized heuristic policy. Those methods necessitate to first estimate a demand distribution and then perform the optimization step (Qi et al. 2020). The common pitfalls are an ill-chosen demand model, the potential of error accumulation during the multiple steps, and often high computational effort.

To the best of our knowledge, the only currently available E2E supervised learning approach for the MPIP has been put forward by Qi et al. (2020). Given historical data, the method learns order quantities. As those depend on the current inventory level and in turn affect future inventory, this approach has to take care of intertemporal dependencies, e.g. by artificially enriching the data to expose the ML model to multiple combinations of the same feature vector with different inventory levels. This complicates the learning process.

In this paper, we aim to solve the above shortcomings by proposing a distribution-free, E2E approach that leverages prior knowledge about a well-performing policy and can predict optimal replenishment decisions in real-time. The core distinction of our method is that it seeks to learn policy parameters, rather than order quantities. We demonstrate our approach in an environment where an (s, S)-policy is provably optimal (see the following two section for details). The concept can be generalized to any setting in which a well-performing policy is known. This is true for many practically relevant settings, and the respective policies often outperform even sophisticated ML approaches (Gijsbrechts et al. 2019). Our method is validated on synthetic data, on which we compare its performance to the ex-post optimal solution as well as a sophisticated model-based, two-step approach.

The remainder of the paper is organized as follows. In Section 2, we present the problem setting and the nature of its optimal solution. Section 3 gives an overview of related work, while our approach is described in detail in Section 4. We describe the setting in which we carried out our experiments in Section 5 and report the results in Section 6. Finally, Section 7 discusses the results, potential limitations and future work.

Multi-Period Inventory Management

The problem that lies at the core of this paper is the multi-period replenishment model with fixed cost, zero lead time, backlogged excess demand and linear holding and backlogging costs. We focus on a finite, but long planning horizon, such that end-of-period effects can be neglected. Mathematically, one may formulate the problem as follows (Scarf 1960; Snyder and Shen 2019).

Let $V_t(I_{t-1})$ denote the minimum expected cost incurred in periods $t, t+1, ..., T$ if we enter period t with starting inventory I_{t-1} and act optimally thereafter. Then the decision in period t is whether to order anything at all and, if so, what level $y > I_{t-1}$ to raise inventory to, and this decision depends on the current inventory, and both the current and future demand. More precisely,

$$\begin{aligned} V_t(I_{t-1}) &= \min_{y \geq I_{t-1}} \big(K\mathbb{1}_{y>I} + c_t(y - I_{t-1}) + L_t(y) \\ &\qquad + \mathbb{E}[V_{t+1}(y - D_t)] \big) \\ &= \min_{y \geq I_{t-1}} \big(G_t(y) + K\mathbb{1}_{y>I_{t-1}} - c_t I_{t-1} \big) , \\ I_t &= y - D_t, \end{aligned} \tag{1}$$

where $z^+ = \max\{z, 0\}$. For future reference we simplified the notation defining $L_t(y) = \mathbb{E}\left[h_t(y - D_t)^+ + b_t(D_t - y)^+\right]$, and $G_t(y) = c_t y + L_t(y) + \mathbb{E}\left[V_{t+1}(y - D_t)\right]$. The notation is clarified further in Table 1. The initial inventory I_0 is fixed externally, and we set $V_{T+1} \equiv 0$.

We emphasize that I_t is allowed to be negative, hence excess demand is backlogged, and we make the non-restrictive assumption that demand is universally bounded, i.e. $|D_t| \leq D$ a.s. for all $t = 1, ..., T$.

Variable Name	Variable Description
T	number of periods
K	fixed order cost
c_t	unit variable order cost in period t
h_t	unit inventory holding cost in period t
b_t	unit backlogging cost in period t
I_t	inventory at the end of period t
D_t	random demand in period t
d_t	realized demand in period t
x_t	feature vector in period t

Table 1: Description of variables.

Under these circumstances, it is well-known that the optimal order policy is of the type $(s_t, S_t)_{t=1,...,T}$, i.e. if $I_{t-1} \leq s_t$ it is optimal to order up to S_t, while for $I_{t-1} > s_t$ we do not place an order at all (Scarf 1960). We emphasize that this result holds for any sequence of potentially non-stationary distributions $D_t \sim \mathcal{D}_t$, and thus it is still true in the case in which those distributions depend on feature data $(x_t)_{t=1,...,T}$, i.e. $D_t \sim \mathcal{D}|x_t$. We exploit this fact to predict the optimal replenishment policy (s_t, S_t) of period t, given the period's feature vector x_t.

Policy Learning

Besides historical demand data $(d_t)_{t=1,...,T}$ we assume availability of contextual information in the form of feature vectors $(x_t)_{t=1,...,T} \subseteq \mathbb{R}^{n \times T}$. Our goal is to learn the parameters (s_t, S_t) of an (s, S) policy, which necessitates labelling the feature data, i.e. assigning a target (s_t, S_t) to each

feature vector x_t.
Thanks to Scarf (1960), we know that

(i) an (s_t, S_t)-policy is optimal in period t, and
(ii) the optimal policy parameters are given by

$$\begin{aligned} S_t &= \operatorname{argmin}_y G_t(y), \qquad \text{and} \\ s_t &= \max\{y \le S_t : G_t(S_t) + K = G_t(y)\}. \end{aligned} \tag{2}$$

Exploiting (i), and in line with Ban (2020), we may reformulate (1) as

$$V_t(I_{t-1}) = \begin{cases} G_t(S_t) + K - c_t I_{t-1}, & I_{t-1} \le s_t, \\ G_t(I_{t-1}) - c_t I_{t-1}, & I_{t-1} > s_t. \end{cases} \tag{3}$$

Plugging (3) into the definition of G_t, we conclude

$$G_t(y) = \begin{cases} c_t y + L_t(y), & t = T, \\ (c_t - c_{t+1})y + c_{t+1}\mathbb{E}[D_t] + L_t(y) & \\ \quad + (G_{t+1}(S_{t+1}) + K)\,\mathbb{E}\left[\mathbb{1}_{y-D_t \le s_{t+1}}\right] & \\ \quad + \mathbb{E}\left[G_{t+1}(y - D_t)\mathbb{1}_{y-D_t > s_{t+1}}\right], & t < T. \end{cases} \tag{4}$$

As a side note for later reference we remark that, for uncensored demand data, Ban (2020) showed that the intuitive way of approximating the expectations on the RHS with empirical means leads to consistent estimators for s_t and S_t. That is, if the distribution of the demand at time t can be sampled from, then sampling N values $d_{t,1}, ..., d_{t,N}$, replacing the expectations in (4) with empirical means over $d_{t,1}, ..., d_{t,N}$ and solving (2) gives policy parameters $(\hat{s}_{t,N}, \hat{S}_{t,N})$ which converge to the true values (s_t, S_t) in probability as $N \to \infty$.

For now, however, our focus is a different one: we wish to label the feature vectors x_t with the ex-post optimal (s_t, S_t)-strategy. That is, we wish to compute the reorder point s_t and up-to-order level S_t that are optimal given $d_t, d_{t+1}, ..., d_T$. We find these ex-post optimal values by plugging the historical demand data $(d_t)_{t=1,...,T}$ into (4), thereby virtually assuming a degenerate, deterministic distribution, and then solve (2) for the ex-post optimal values (s_t, S_t). Independent of the yet unknown starting inventory I_{t-1}, these values represent the policy for period t that is optimal in the long-term, assuming that demand unfolds exactly as $d_t, d_{t+1}, ..., d_T$.

Solving (2) entails four technical steps. For given G_{t+1}, we first pre-compile G_t based on a fine-grid approximation using cubic splines. Second, we compute initial guesses for (s_t, S_t) using the algorithm put forward by Bollapragada and Morton with a modified step size of .5 (section 2.2 in Bollapragada and Morton, 1999). Third, we solve (2) using scipy's implementation of the BFGS algorithm (Broyden 1970; Fletcher 1970; Goldfarb 1970; Shanno 1970) to find S_t. Fourth, we use Brent's method (Brent 1973) to solve the root problem $G_t(S_t) + K = G_t(s_t)$. Finally, we label feature vector x_t by the 2-dimensional target (s_t, S_t).

Upon completion of the labelling step, we may train any supervised learning algorithm on the labelled data $(x_t(s_t, S_t))_{t=1,...,T}$. In essence, given a feature vector x_t, this algorithm learns to predict demand for a number of periods m ahead, to assemble the corresponding functions $G_t, ..., G_{t+m}$, and then to recursively solve (2) to arrive at the optimal policy values (s_t, S_t). We note that while in theory $m = T - t$, we observed in our experiments that after a small number of periods m, the effect of future periods on the current optimal policy values vanishes. We established empirically that for our parameters, $m \approx 20$.
Due to the complexity of the function to be learned, the fact that deep neural networks (DNNs) are powerful enough to learn any continuous function (Hornik 1991), and as DNNs have been applied successfully in time series forecasting (Zhang, Eddy Patuwo, and Y. Hu 1998; Oroojlooyjadid, Snyder, and Takác 2016; Müller et al. 2020), we opt for a DNN as our choice of supervised learning algorithm. Henceforth, we refer to our method as "(s, S)-DNN".

Experimental Setting

We test the performance of (s, S)-DNN with respect to two metrics, total cost and computation time required for prediction. For that purpose, we require a data set comprising feature and demand data, which we then have to label by the optimal policy parameters (s_t, S_t). To highlight the importance of feature data, we simulate demand as a function of generated features subject to some additive as well as multiplicative, i.e. heteroscedastic noise. Heteroscedasticity induces the demand distribution to depend even stronger and more subtly on the auxiliary data, which is hard to pick up for feature-ignorant approaches. To demonstrate the significance of our findings, we create multiple data sets and report performance statistics such as median and confidence intervals. Creation of both feature and demand data were inspired by Bertsimas and McCord (2019).

Specifically, we create $i = 1, ..., N = 100$ i.i.d. data sets, each containing $t = 1, ..., T = 640$ $(x_{t,i}, d_{t,i})$ records with $x_{t,i} \in \mathbb{R}^5, d_{t,i} \in \mathbb{R}_{\ge 0}$. The 5 features evolve as independent (p, q)-ARMA time series, where $1 \le p^j \le 2$ and $0 \le q^j \le 2$ for $j = 1, ..., 5$. More precisely, we have

$$x_{t,i}^j = \varepsilon_{t,i}^j + \sum_{l=1}^{p^j} a_l^j x_{t-l,i}^j + \sum_{l=1}^{q^j} b_l^j \varepsilon_{t-l,i}^j$$

for $j = 1, ..., 5$ with $x_{t,i}^j = \varepsilon_{t,i}^j = 0$ for all i, j and $t < 0$ and $\varepsilon_{t,i}^j \sim \mathcal{N}(0, 1)$ otherwise . Demand is generated following a simple factor model, i.e.

$$d_{t,i} = \max\{0, 250 + f_i^T x_{t,i} + h_i^T x_{t,i}\theta_{t,i} + \psi_{t,i}.\}$$

We set $f = [17, 22, -12, 5, -8]^T, h = [1, 0, 0, 0, 0]^T$ and let f_i, h_i be arbitrary permutations thereof, distinct across data sets but fixed for a given set. Noise is governed by $\theta_{t,i} \sim \mathcal{N}(0, 125)$, and $\psi_{t,i} \sim \mathcal{N}(0, 50)$.

For simplicity, cost factors were held constant over time at $K = 500, c = 1.5, h = 1, b = 5$. For each data set, the labels $(s_{t,i}, S_{t,i})$ were computed for a training set of 520 periods and a test set of 120 periods separately. The last 20 periods of both sets were discarded to avoid any end-of-period effect, as foreshadowed in the Section "Policy Learning". For each data set, we train a deep neural network on a training set of 500 $(x_t, (s_t, S_t))$ records and subsequently evaluate it on a test horizon of 100 periods.

Although the underlying dynamics for feature and demand data were the same across data sets, performance could be improved by searching for individual hyperparameters. That is, for each dataset we optimized the number of layers L – not counting the input layer – ($2 \leq L \leq 4$), the number of epochs E ($500 \leq E \leq 6000$), the learning rate lr ($0.001 \leq lr \leq 0.2$), the number of nodes per layer n_l ($n_l \in [2^3, 2^4, 2^5, 2^6, 2^7]$ for all $l = 1, ..., L$), and the regularization parameters ("dropout" vs. "L2", while for the latter we chose $100 \leq \lambda \leq 1000$). We sped up the hyperparameter search resorting to sequential model-based optimization using Bergstra, Yamins, and Cox's (2013) tree parzen estimator method. We experimented with both a generic RMSE and a problem-specific loss function. The latter considers the fact that deviations from the target (s_t, S_t) have asymmetric implications in two ways. First, on average, slight deviations from the reorder point s_t can be expected to have a smaller effect than deviations of the same magnitude from the up-to-order level S_t, as the latter induces incorrect order quantities in every period with a non-zero order quantity, while the first has only an effect if a period's starting inventory I_{t-1} is close to s_t. Second, for both reorder point and up-to-order level, depending on the cost structure over- and underestimating the optimal parameter can have different implications, and thus they should be punished differently. Performance was improved further when trained with this problem-specific loss. The exact problem-specific loss function is given by

$$L((\hat{s}_t, \hat{S}_t, (s_t, S_t)) = C_S^+(\hat{S}_t - S_t)^+ + C_S^-(S_t - \hat{S}_t)^+ \\ + C_s^+(\hat{s}_t - s_t)^+ + C_s^-(s_t - \hat{s}_t)^+.$$

The coefficients $C_S^+, C_S^-, C_s^+, C_s^-$ express the expected effect of a unit deviation from the respective component in the respective direction assuming we act optimally in all other periods. In detail, we define $C_t(I, d) = h_t(I-d)^+ + b_t(d-I)^+$ and compute

$$C_S^+ = \frac{1}{T-k} \sum_{t=1}^{T-k} C_{t,k}(s_t, S_t{+}1, I_{t-1}) - C_{t,k}(s_t, S_t, I_{t-1}), \tag{5}$$

with

$$\begin{aligned} C_{t,k}(s, S, I) = {} & [K + c_t(S - I) + C_t(S, d_t)]\, \mathbb{1}_{I \leq s} \\ & + C_t(I, d_t) \mathbb{1}_{I > s} \\ & + \sum_{i=t+1}^{t+k} \Big([K + c_i(S_i - I_{i-1}) + C_i(S_i, d_i)]\, \mathbb{1}_{I_{i-1} \leq s_i} \\ & + C_i(I_{i-1}, d_i) \mathbb{1}_{I_{i-1} > s_i} \Big), \\ I_i = {} & I_{i-1} - d_i + (S_i - I_{i-1}) \mathbb{1}_{I_{i-1} \leq s_i}, \text{ and} \\ I_{t-1} = {} & I, S_t = S, s_t = s, \end{aligned}$$

where I_{t-1} is the ending inventory of period $t-1$ if we act optimally up to and including period $t-1$. C_S^-, C_s^+, and C_s^- are estimated analogously. The parameter k needs to be chosen carefully such that $T-k$ is small enough for (5) to be a good approximation of the expectation, but large enough for (5) to capture the full cost effect of a deviation. An empirical analysis showed that any value $20 \leq k \leq 100$ led to similar, stable results. We chose $k = 100$.

To assess the performance of (s, S)-DNN on the test set, we compare it to two baselines. First, we compute the ex-post optimal solution of (1) by plugging in the historical demand values d_t for the random variables D_t. The resulting mixed integer linear program (MILP) is a modification of the Wagner-Whitin model (Snyder and Shen, 2019, Ch. 3) to allow for backorders:

$$\begin{aligned} \min_{q_1,...,q_T} \quad & \sum_{t=1}^{T} KS_t + c_t q_t + h_t o_t + b_t u_t \\ & q_t \leq MS_t, \qquad t = 1, ..., T, \\ & I_t = I_{t-1} + q_t - d_t, \qquad t = 1, ..., T, \\ & o_t \geq I_t, \qquad t = 1, ..., T, \\ & u_t \geq -I_t, \qquad t = 1, ..., T, \\ & S_t \in \{0, 1\}, q_t, u_t, o_t \geq 0, \qquad t = 1, ..., T, \end{aligned} \tag{6}$$

where M denotes a large constant satisfying $\max\{q_t^* | t = 1, ..., T\} \leq M$ for the optimal order quantities $(q_t^*)_{t=1,...,T}$. We solved (6) with Gurobi 9.1.2 to optimality.

The second approach, (s, S)-logistic, is a classic model-based, two-step approach. We first choose a parameterized distribution as demand model and estimate its parameters from historical demand data. We then perform the optimization step by simulating demand data for the test horizon and solving (2) with G_t given by (4). Ban (2020) showed that for only 50 demand values per period, 95% of the relative absolute errors were smaller than 2% and difference in expected cost was 0.6%. Based on these fast convergence properties, we decided to simulate 50 demand values per period. Demand was tested for several distributions using the Kolmogorov-Smirnov test, with the logistic distribution turning out to be an extraordinarily good fit. At the 1% significance level, there is not a single data set for which the null hypothesis that demand follows a logistic distribution is rejected (cf. Figures 1 and 2). We note that this results in very high expectations on the cost performance of this approach. If demand was truly logistically distributed then asymptotically speaking, the resulting (s_t, S_t) policy would indeed be optimal – based on our distribution tests and Ban's analysis, both of those assumptions are realistic.

Results

Our most important findings are summarized in Figure 3. On average, we can observe that the (s, S)-DNN method operates within a 10.85% gap to the ex-post optimal solution (MILP) and outperforms the model-based approach (s, S)-logistic w.r.t. total cost by 2.41% (Figure 3 a)). This is particularly noteworthy given the excellent fit of the logistic distribution, which causes (s, S)-logistic to perform extraordinarily well. Furthermore, (s, S)-DNN can make predictions in real-time and thus saves a considerable amount of effort and computation time in comparison to a multi-step approach (Figure 3 b)).

To allow a more detailed analysis, we introduce the fol-

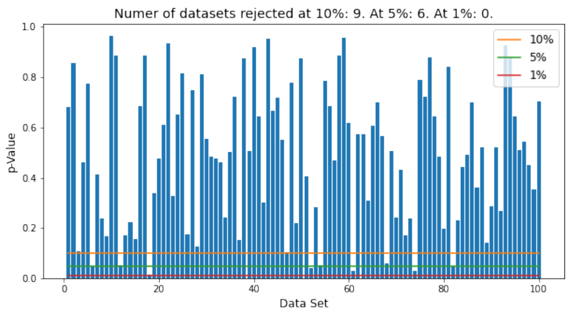

Figure 1: Testing the demand data for logistic distribution - p-values of Kolmogorov-Smirnov tests by data set.

lowing taxonomy, characterizing any approach solving (1) along four binary dimensions:

(a) Use of prior knowledge: yes vs. no

(b) Distribution approximation: non-parametric (NP) vs. model-based (MB)

(c) Use of feature data: yes vs. no

(d) Number of steps: E2E vs. multi-step

Our own approach, (s,S)-DNN, constitutes a non-parametric, feature-aware, E2E method that manages to leverage knowledge about the nature of the optimal solution. On the other hand, (s,S)-logistic can be categorized as a model-based, feature-ignorant, multi-step method that is aware of the optimal solution structure. While we can venture educated guesses about the performance effect of some of the components (e.g., it is fairly straightforward that multi-step methods require more computational effort at prediction time), in general it is not obvious to which extent the individual characteristics contribute to an approach's performance. To investigate the isolated effect of a single component, we need to carry out experiments which compare approaches that differ *in only that component*. As we cannot simply turn off/on every individual component in our existing approaches (s,S)-DNN and (s,S)-logistic, it is necessary to develop new methods. We opt for the following three additional approaches, which all constitute slight modifications of (s,S)-DNN or (s,S)-logistic that mimic their behaviour if only one characteristic were modified:

(i) (s,Q)-logistic: In order to analyze the effect of leveraging prior knowledge about the optimal policy structure, we introduce a model-based, feature-ignorant, multi-step method that mimics the performance of (s,S)-logistic if the policy was mis-specified as an (s,Q) policy. An (s_t,Q_t) policy orders quantity Q_t whenever the starting inventory I_{t-1} drops below the reorder point s_t, instead of raising it to some up-to-order level S_t. We choose (s,Q) as opposed to other policies (e.g., base-stock and capped base stock) as it is widely used and performed the best among the mentioned misspecified strategies on the development set. To determine s_t and Q_t, we simulate logistically distributed demand samples with dynamically updated parameters, select Q_t as the economic order quantity and s_t – to maintain comparability – as the quantile of the logistic distribution that matches the cycle service level obtained by (s,S)-logistic. This allows us measure the effect of specifying the crrect policy by comparing (s,Q)-logistic to (s,S)-logistic.

(ii) (s,S)-NP: To measure the impact of approximating the demand non-parametrically instead of via an inflexible distribution model, we suggest to monitor (s,S)-NP, a model-free, feature-ignorant, multi-step method that leverages prior knowledge and thus mimics the behaviour of (s,S)-logistic if we simply changed the latter's way of modelling demand. Consequently, instead of sampling from a logistic distribution, we simulate $d_{t,1},...,d_{t,N}$ with $N = 50$ for the entire test set $(t = 1,...,120)$ by sampling with replacement from the historical demand. Subsequently, we solve (2) in the exact same way as we did for (s,S)-logistic. This allows us to measure the per-

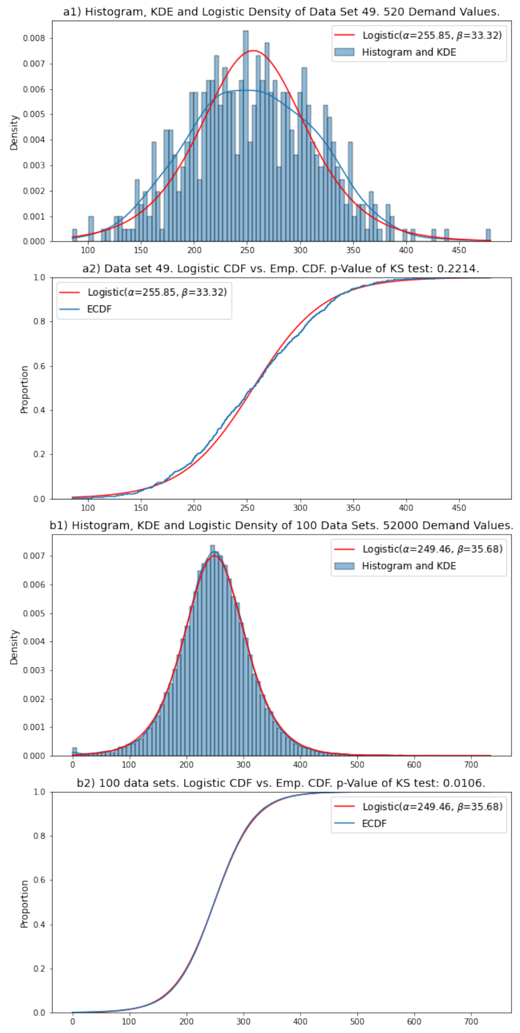

Figure 2: Visual goodness-of-fit test of demand data during training for logistic distribution. Subplots a1) and a2) refer to data from one randomly chosen data set, subplots b1) and a2) to data from all 100 sets.

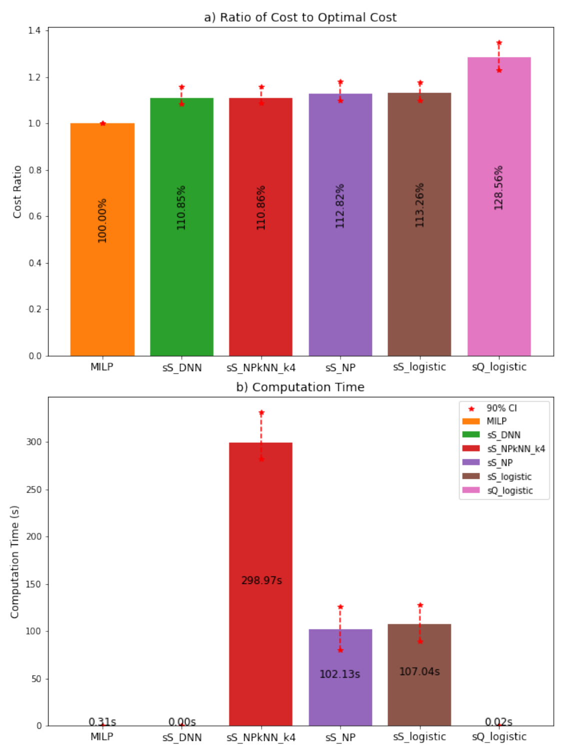

Figure 3: Performance metrics. a) Ratio of cost to ex-post-optimal cost (MILP). b) Computation time for predictions on test horizon (100 periods) in seconds. Median and 90% confidence intervals over 100 data sets.

formance effects of the way we approximate the demand distribution by comparing (s, S)-NP to (s, S)-logistic.

(iii) (s, S)-NPkNN: In an attempt to evaluate the effect of leveraging auxiliary data and develop a method that is comparable to both (s, S)-NP and (s, S)-DNN, we refine (s, S)-NP in the following way. From our earlier analysis, we know that the current period's optimal policy values (s_t, S_t) are affected by only the current and the future 19 periods. For a given feature vector x_t, we select k distinct nearest neighbors $x_{s_1}, ..., x_{s_k}$ with $s_k \leq t - 20$ and approximate relevant future demand by $d_{t+j,i} = d_{s_i+j}$, $j = 0, 1, ..., 19$, $i = 1, ..., k$. We determined $k = 4$ by cross-validation on the development set. Finally, we solve (2) in the usual way. This constitutes a non-parametric, feature-aware, multi-step method that leverages prior knowledge. It differs from (s, S)-NP only by the treatment of feature data. Furthermore, it mimics the behaviour of (s, S)-DNN with the only difference that the optimization step is performed explicitly as opposed to in an integrated way. Therefore, (s, S)-NPkNN qualifies for analyzing the effect of feature data (when compared to (s, S)-NP) and multi-step integration (when compared to (s, S)-DNN).

The characteristics of all of the approaches are summarized in Table 2. In the following, we report the results of the four experiments. This can be viewed as an in-

Table 2: Taxonomy of MPIP approaches. The row ”Focus” refers to the component an approach is used to analyse, o) being the original comparison (MILP vs. (s, S)-DNN vs. (s, S)-logistic).

	MILP	(s, Q)-logistic	(s, S)-logistic	(s, S)-NP	(s, S)-NPkNN	(s, S)-DNN
a) Prior Knowledge	-	No	Yes	Yes	Yes	Yes
b) Distribution	-	MB	MB	NP	NP	NP
c) Feature Data	-	No	No	No	Yes	Yes
d) # Steps	-	Multi-Step	Multi-Step	Multi-Step	Multi-Step	E2E
Focus	o)	a)	o), a), b)	b), c)	c), d)	o), d)

cremental refinement of the most naive approach (s, Q)-logistic by first leveraging prior knowledge ((s, Q)-logistic $\rightarrow$ (s, S)-logistic), secondly relaxing the distribution assumption ((s, S)-logistic $\rightarrow$ (s, S)-NP), thirdly including auxiliary data ((s, S)-NP $\rightarrow$ (s, S)-NPkNN) and finally integrating the multi-step procedure into an E2E method to arrive at the most sophisticated approach, (s, S)-DNN. The results are summarized in Figure 3. Performance differences between two approaches were tested for significance using a paired t-test.

(a) Prior knowledge: (s, Q)-logistic vs. (s, S)-logistic. While (s, Q)-logistic optimizes its parameters using a single-period function, (s, S)-logistic requires a dynamic program to be solved. It is therefore not surprising that the latter is computationally more expensive. We emphasize the cost difference of 15.3% and the large optimality gap of (s, Q)-logistic (28.56%), both of which are significant at the 1% level. This demonstrates the importance of considering prior knowledge about the structure of the optimal strategy.

(b) Distribution approximation: (s, S)-logistic vs. (s, S)-NP. As both methods solve the identical optimization problem of the same complexity ($N = 50$ for both methods), the computational effort is comparable. Given the good fit of the logistic distribution, relaxing the logistic assumption does not yield a noticeable cost improvement – with a p-value of 0.42, the decrease in median cost of 0.44% was not significant. This can be expected to change when repeating the experiment with real-world data, as it is unlikely for real data to follow a specific distribution as closely.

(c) Feature data: (s, S)-NP vs. (s, S)-NPkNN. The noticeable and significant difference in the computation time comes from the fact that (s, S)-NP solves the dynamic optimization problem of recursively computing (s_t, S_t) once for a horizon of 100 periods, while (s, S)-NPkNN repeats the procedure for each of the 100 periods of the test horizon, each time solving the optimization problem with a 20 period horizon. In effect, this means that (s, S)-NPkNN solves a problem with 2000 periods. It is only due to the difference in complexity of the problems ($N = 50$ for (s, S)-NP, $N = 4$ for (s, S)-NPkNN) that this does not lead to a time increase of factor 20. The higher computational effort allows an effective use of feature data that generates comparability of (s, S)-NPkNN and (s, S)-DNN, as the latter approach leverages auxiliary data of every period, too. In an attempt to increase comparability to (s, S)-NP by aligning the computation times, one could have added an approach (s, S)-NPkNN-naive, which generates sequences of length 100 instead of 20 and thereby only makes use of the test horizon's initial feature vector. We avoided this for simplicity. The median cost difference of 1.96% between the two approaches was significant at the 1% level and demonstrates the importance of considering feature data, as well as using it in an effective way.

(d) Number of steps: (s, S)-NPkNN vs. (s, S)-DNN. The most obvious difference between the two methods is of course the reduction of computational effort by integrating multiple steps into one. We note that this is of practical interest in itself, as especially an approach as (s, S)-NPkNN is hardly applicable in a practical environment, where computational complexity is increased even further by the necessity of a larger k due to a more complex feature-demand-dependence and the presence of multiple products. With this in mind, it is an achievement in itself to efficiently integrate the two complex steps of prediction and optimization into one single function without suffering a considerable cost increase. (s, S)-DNN achieves this task without incurring any cost increase. With a p-value of .95, the actual median cost decrease of .01% was not significant - it is safe to say that the two approaches perform on the same level. Due to the complexity of the function the DNN has to learn and the excellent demand approximation of (s, S)-NPkNN, the finding that a sophisticated multiple-steps method can compete with an E2E approach is not surprising, and it is in line with results presented in Punia, Singh, and Madaan (2020) and Müller et al. (2020), where multi-step methods have even been shown to outperform integrated methods for specific tasks.

Discussion & Future Work

On average, our flagship E2E ML Method (s, S)-DNN performed within a 10.85% gap of the ex-post optimal solution and outperformed the standard multi-step, model-based method (s, S)-logistic by 2.41%, leading to a cost decrease that was significant at the 1% level. This demonstrates the learnability of optimal ordering policies to a degree where an ML method can outperform computationally more expensive multi-step, model-based methods. This is especially noteworthy as the specified model fits the data exceptionally well and the computational effort guarantees a close-to optimal solution.

While any E2E ML approach may require considerable initial effort to search for optimal hyperparameters and to train the model, this can be made up for at prediction time. When put to production in a real-world environment, this can provide substantial efficiency gains.

Judging from the detailed analysis of the approaches ' individual characteristics, the performance effects can be ranked in the following order. The most important notion is to take domain knowledge into account, even when resorting to an ML solution. This confirms the result of Gijsbrechts et al. (2019), where, despite considerable training effort, an RL agent that ignores potential knowledge about well-performing policies was outperformed by a simple heuristic, a capped base-stock policy. Secondly, we demonstrated that the effective use of meaningful auxiliary data can boost performance. Thirdly, we want to highlight that for our specific data, replacing an inflexible model with a non-parametric way of approximating the demand distribution did yield a slight but insignificant performance improvement. This can be attributed to the unrealistically good fit of the logistic distribution which is unlikely to repeat itself in an experiment with real-world data. Finally, encapsulating demand prediction and the complex optimization step into a single function achieved large efficiency gains while maintaining the same level of performance.

As a consequence, we interpret our initial results as a promising start into the field of learning ordering policies in a supervised fashion and plan to refine our work by the following steps. First of all, we are confident that the deep neural network of our (s, S)-DNN approach can improve further. So far, (s, S)-DNN has been trained with increasingly sophisticated methods (data set-individual hyperparameters, sequential model-based hyperparameter optimization, problem-specific loss function), but rather low computational power, suggesting an increase of raw computing time as an obvious next step. Secondly, we realize that blatantly misspecifying the ordering policy is a rather aggressive way of demonstrating the importance of leveraging domain knowledge. While the chosen (s, Q)-policy is justifiable due to its popularity and superior performance among several heuristics, we plan to replace it by a policy-free reinforcement learning approach. We hope this lets us emphasize the importance of considering prior knowledge even more strongly. Thirdly, while our experiments on synthetic data serve as a proof-of-concept, we have yet to demonstrate the soundness of (s, S)-DNN on real-world data. While the observed challenges of integrating prediction and optimization into one step will surely remain, we expect the effect of approximating the demand distribution non-parametrically to play a much higher role and the positive impacts of leveraging prior knowledge and auxiliary data to repeat. This gives us confidence that (s, S)-DNN yields good results on real-world data. Finally, we plan to investigate other problem settings, such as extensions with positive or stochastic lead times and multiple products. In many of those environments, well-performing policies exist, even if they are not proven to be optimal. It will be interesting to see if and with how much effort we can transfer our approach to such problems, and what level of performance can be achieved.

References

Arrow, K. J.; Harris, T.; and Marschak, J. 1951. Optimal Inventory Policy. *Econometrica*, 19(3): 250–272.

Ban, G.; and Rudin, C. 2018. The Big Data Newsvendor: Practical Insights from Machine Learning. *Operations Research*, 67(1): 90–108.

Ban, G.-Y. 2020. Confidence Intervals for Data-Driven Inventory Policies with Demand Censoring. *Oper. Res.*, 68(2): 309–326.

Bergstra, J.; Yamins, D.; and Cox, D. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 115–123. Atlanta, Georgia, USA: PMLR.

Bertsimas, D.; and McCord, C. 2019. From Predictions to Prescriptions in Multistage Optimization Problems. arXiv:1904.11637.

Bollapragada, S.; and Morton, T. E. 1999. A Simple Heuristic for Computing Nonstationary (s, S) Policies. *Operations Research*, 47(4): 576–584.

Brent, R. P. 1973. *Algorithms for Minimization Without Derivatives*, chapter 3-4. Englewood Cliffs, NJ: Prentice Hall.

Broyden, C. G. 1970. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1): 76–90.

Ehrhardt, R. 1984. (s, S) Policies for a Dynamic Inventory Model with Stochastic Lead Times. *Operations Research*, 32(1): 121–132.

Fletcher, R. 1970. A new approach to variable metric algorithms. *The Computer Journal*, 13(3): 317–322.

Gijsbrechts, J.; Boute, R.; Zhang, D.; and Van Mieghem, J. 2019. Can Deep Reinforcement Learning Improve Inventory Management? Performance on Dual Sourcing, Lost Sales and Multi-Echelon Problems. *SSRN Electronic Journal*.

Goldfarb, D. 1970. A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation*, 24(109): 23–26.

Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2): 251–257.

Huber, J.; Müller, S.; Fleischmann, M.; and Stuckenschmidt, H. 2019. A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278(3): 904–915.

Kaplan, R. S. 1970. A Dynamic Inventory Model with Stochastic Lead Times. *Management Science*, 16(7): 491–507.

Müller, S.; Huber, J.; Fleischmann, M.; and Stuckenschmidt, H. 2020. Data-driven Inventory Management under Customer Substitution. *https://papers.ssrn.com/*.

Oroojlooyjadid, A.; Snyder, L. V.; and Takác, M. 2016. Applying Deep Learning to the Newsvendor Problem. *CoRR*, abs/1607.02177.

Punia, S.; Singh, S. P.; and Madaan, J. 2020. From Predictive to Prescriptive Analytics: A Data-Driven Multi-Item Newsvendor Model. *Decision Support Systems*, 136.

Qi, M.; Mak, H.-Y.; and Shen, M. 2020. Data-driven research in retail operations—A review. *Naval Research Logistics (NRL)*, 67.

Qi, M.; Shi, Y.; Qi, Y.; Ma, C.; Yuan, R.; Wu, D.; and Shen, Z.-J. M. 2020. A Practical End-to-End Inventory Management Model with Deep Learning. *Social Science Research Network*.

Scarf, H. 1960. The Optimality of (S, s) Policies in the Dynamic Inventory Problem. *Math Methods Soc Sci*.

Shanno, D. F. 1970. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, 24(111): 647–656.

Snyder, L. V.; and Shen, Z.-J. M. 2019. *Fundamentals of Supply Chain Theory*. John Wiley & Sons, Ltd. ISBN 9781119584445.

Zhang, G.; Eddy Patuwo, B.; and Y. Hu, M. 1998. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1): 35–62.