ORIGINAL ARTICLE

POMS

Addressing distributional shifts in operations management: The case of order fulfillment in customized production

Julian Senoner[1] | **Bernhard Kratzwald**[1] | **Milan Kuzmanovic**[1] | **Torbjørn H. Netland**[1] | **Stefan Feuerriegel**[2]

[1]ETH Zurich, Department of Management, Technology and Economics, Zurich, Switzerland

[2]LMU Munich, Institute of Artificial Intelligence (AI) in Management, Munich, Germany

Correspondence
Julian Senoner, ETH Zurich. Department of Management, Technology, and Economics. 8092 Zurich, Switzerland.
Email: jsenoner@ethz.ch

Handling Editor: Panos Kouvelis

Abstract

To meet order fulfillment targets, manufacturers seek to optimize production schedules. Machine learning can support this objective by predicting throughput times on production lines given order specifications. However, this is challenging when manufacturers produce customized products because customization often leads to changes in the probability distribution of operational data—so-called *distributional shifts*. Distributional shifts can harm the performance of predictive models when deployed to future customer orders with new specifications. The literature provides limited advice on how such distributional shifts can be addressed in operations management. Here, we propose a data-driven approach based on adversarial learning, which allows us to account for distributional shifts in manufacturing settings with high degrees of product customization. We empirically validate our proposed approach using real-world data from a job shop production that supplies large metal components to an oil platform construction yard. Across an extensive series of numerical experiments, we find that our adversarial learning approach outperforms common baselines. Overall, this paper shows how production managers can improve their decision making under distributional shifts.

KEYWORDS
adversarial learning, distributional shifts, machine learning, manufacturing, order fulfillment

1 | INTRODUCTION

Order fulfillment is a crucial performance indicator in manufacturing (Song et al., 1999). Achieving on-time delivery can be particularly difficult when manufacturers produce highly customized products (Cohen et al., 2003). For these manufacturers, incoming customer orders can involve multiple nonstandard production tasks that must be planned effectively. This is challenging because order throughput times vary across customer specifications. Completing production orders too early causes unnecessary inventory costs and increases the risk of rework in case of engineering changes, whereas delays diminish service levels and can lead to substantial economic losses (Cohen et al., 2003). Therefore,

reducing deviations from order delivery due dates is important for maintaining a good cost performance. Motivated by this objective, manufacturers continuously seek to improve their planning accuracy.

Machine learning can support manufacturers in achieving high planning accuracy. Specifically, machine learning can improve managerial decision making by utilizing historical observations to predict throughput times given order specifications (cf. Grabenstetter & Usher, 2014). This enables production managers to optimize their production plans and reduce costs by mitigating deviations from order delivery due dates. However, implementing machine learning in manufacturing settings with high degrees of product customization is challenging because orders may involve unique specifications and high variety (e.g., producing customized components for one-of-a-kind oil platforms). Due to heterogeneity between different customer orders, such settings typically generate

---

Accepted by Panos Kouvelis, after three revisions.

operational data for which probability distributions of different customer orders are highly dissimilar. In the machine learning literature, this is commonly referred to as *distributional shifts*. Consequently, the standard assumption of identically distributed samples in predictive analytics (Hastie et al., 2009) is violated, and the prediction performance may deteriorate.

Scholars have argued that distributional shifts are a key challenge of successfully applying predictive analytics in management (Simester et al., 2020). While there is increasing traction of predictive analytics in operations management (OM) (e.g., Baardman et al., 2018; Bastani et al., 2022; Mišić & Perakis, 2020; Olsen & Tomlin, 2020), there is, however, scarce advice on how distributional shifts can be addressed. Existing approaches to address distributional shifts make use of model retraining (Cui et al., 2018) or transfer learning via fine-tuning (Bastani, 2021; Kouw & Loog, 2018; McNamara & Balcan, 2017; Pan & Yang, 2010). These methods are effective in make-to-stock manufacturing (e.g., fast-moving consumer goods) characterized by low variety and high volumes. Yet, they are likely to fall short in manufacturing settings characterized by high variety and low volumes. Both model retraining and transfer learning via fine-tuning require labeled data for both historical and forthcoming orders, yet labels for the latter are not available when dealing with new orders. As such, tailored approaches for customized production are needed.

To meet order fulfillment targets in customized production, we develop a data-driven approach to predict order throughput times and then perform job shop scheduling. Key to our approach is that we predict when the order will be finished and, then, use this information to optimize scheduling decisions. Due to the operational heterogeneity in customized production, making predictions of throughput times is challenging as there are distributional shifts between different customer orders, which violate the standard assumption of machine learning and which can reduce prediction performance. In our approach, we account for distributional shifts between different customer orders through the use of adversarial learning. Adversarial learning (Goodfellow et al., 2014) is a recent innovation in artificial intelligence to make inferences under two opposing—thus adversarial—objectives. In our case, we leverage adversarial learning to combine the following two objectives: (1) to predict throughput times with the best possible performance and, simultaneously, (2) to minimize the distance of the neural network representations of the operational data between the historical and the forthcoming order setting. Importantly, the latter accounts for the new specifications of the forthcoming order setting (without knowing the ground-truth labels) and, as a result, yields better predictions of throughput times and better scheduling decisions for future orders.

We evaluate the effectiveness of our proposed approach in a series of numerical experiments using real-world industrial data from Aker Solutions (from here on Aker), a leading engineering company in the energy sector. We focus on a job shop production that supplies large metal components for the construction of oil platforms. In this case, components produced for different customer order settings (i.e., new oil platforms) involve many idiosyncratic specifications. We find a substantial distributional shift between order settings; that is, the conditions under which the components are produced are highly dissimilar. Especially when starting to produce components with new specifications, it is challenging for naïve predictions using machine learning to provide accurate estimates of throughput times. This is addressed in our approach based on adversarial learning as it explicitly accounts for the distributional shifts between different order settings. We then compare our proposed approach for job shop scheduling against several data-driven baselines. Across an extensive series of numerical experiments, we find that our approach outperforms the baselines and can offer considerable cost savings.

This paper makes three main contributions. First, we provide empirical evidence of how predictive analytics can improve order fulfillment in customized production. Second, we contribute to the practice and literature on predictive analytics in OM (e.g., Bastani et al., 2022; Mišić & Perakis, 2020; Olsen & Tomlin, 2020). Here, we address the problem of distributional shifts, which is particularly pertinent in settings subject to extensive heterogeneity. Third, we tailor adversarial learning to an OM context and study its operational value for job shop scheduling. Altogether, our work has direct managerial implications: Manufacturers need to identify and cater for distributional shifts in customized production. Simply deploying predictive analytics without addressing distributional shifts may result in subpar decisions.

This paper is structured as follows. Section 2 motivates the challenges of predictive analytics in customized production under distributional shifts, yet revealing a scarcity of methods for addressing distributional shifts in OM. Section 3 introduces our empirical setting at Aker. Section 4 proposes our model for job shop scheduling, which uses adversarial learning to predict throughput times while addressing distributional shifts. Section 5 evaluates our approach in a series of numerical experiments under different operational contexts. We report various robustness checks in Section 6. Based on our results, Section 7 discusses the implications for making robust inferences under distributional shifts in OM.

2 | RELATED WORK

This paper is motivated by the practical problem of making robust inferences under distributional shifts and, for this purpose, draws upon statistical learning theory. We see four streams of research as particularly relevant to this work: (1) predictive analytics in OM, (2) job shop scheduling, (3) distributional shifts, and (4) adversarial learning.

2.1 | Predictive analytics in OM

Predictive analytics can support managers in making decisions by modeling uncertain operational outcomes (Choi et al., 2018; Cohen, 2018; Feuerriegel et al., 2022). Recent methodological advances, accompanied by the increasing availability of data, have accelerated the adoption of predictive analytics in OM (Bastani et al., 2022; Jakubik & Feuerriegel, 2022; Mišić & Perakis, 2020; Olsen & Tomlin, 2020). In the following, we provide an overview of predictive analytics in OM. For a detailed review of the literature, see Mišić & Perakis (2020) and Bastani et al. (2022).

There are many promising demonstrations of predictive analytics in OM, such as sales and demand forecasting (Baardman et al., 2018; Carbonneau et al., 2008; Cui et al., 2018; Ferreira et al., 2016; Lau et al., 2018), revenue management (Bernstein et al., 2019; Chen et al., 2022; Feldman et al., 2021), location selection problems (Glaeser et al., 2019; Huang et al., 2019), last mile delivery (Liu et al., 2020), product recall decisions (Mukherjee & Sinha, 2018), inventory management (Bertsimas & Kallus, 2019), procurement under demand uncertainty (Ban et al., 2019), and the estimation of the remaining useful life of products (Mazhar et al., 2007).

In manufacturing operations, Senoner et al. (2022) adapted predictive analytics to improve process quality. However, despite their relevance, applications of predictive analytics and decision making are still scarce in manufacturing. OM scholars have therefore argued that more attention in this "under-researched" area is needed (Feng & Shanthikumar, 2018). This particularly holds true for manufacturing settings, where products are manufactured with high variety and in low volumes. These characteristics make it particularly challenging to apply conventional prediction methods due to distributional shifts. Here, our paper contributes to the OM literature by tailoring predictive analytics for manufacturing settings with high degrees of product customization.

Applying predictive analytics to data-scarce settings (e.g., new products) is a known challenge in OM. Kesavan and Kushwaha (2020) compare analytics and expert decision making in a field experiment, finding that giving discretionary power to experts is beneficial in growth-stage products. Instead of experts, practitioners can also revert to social media (Cui et al., 2018) or other proxies (Bastani, 2021). However, our work is different in that there are no such data available for forthcoming orders in customized production. As a remedy, we develop a data-driven approach based on adversarial learning and later demonstrate its operational value.

2.2 | Job shop scheduling

Job shop scheduling problems consider a multitude of arriving production orders that compete for processing time on common resources (Adams et al., 1988; Wein & Chevalier, 1992). These orders are typically associated with an arrival date t, a due date d, and a stochastic throughput time y.

The difference between d and t defines the planned leadtime $y^{(plan)}$. The problem is to schedule the individual production orders by optimizing against a time-dependent objective. A common objective is to minimize the total cost of (1) *earliness*, that is, completing production orders too early (before d) and (2) *tardiness*, that is, completing production orders too late (after d) (e.g., Atan et al., 2016; Bagchi et al., 1994; Federgruen & Mosheiov, 1996; Seidmann & Smith, 1981). A simple form of the total cost function C is given by

$$C(y, y^{(plan)}) = \zeta^{(early)}[y^{(plan)} - y]^+ + \zeta^{(tardy)}[y - y^{(plan)}]^+,$$

(1)

where $[\cdot]^+$ returns the positive part of an expression and where the convex functions $\zeta^{(early)}(\cdot)$ and $\zeta^{(tardy)}(\cdot)$ measure the costs of earliness and tardiness per time unit (Seidmann & Smith, 1981). As can be seen in Equation (1), the planning accuracy in a job shop production strongly depends on accurate time estimates for throughput times.

The throughput time y is typically estimated based on historical observations. Recent approaches have also included covariate information for the estimation of throughput times (Grabenstetter & Usher, 2014). The common assumption is that data from previous customer orders are sampled from the same probability distribution as the data from forthcoming customer orders. This rarely holds in manufacturing settings with high degrees of product customization. Customization often leads to changes in the distribution of operational data, which makes it particularly challenging to provide accurate throughput time estimates. This paper seeks to predict throughput times while accounting for distributional shifts in the operational data due to product customization.

2.3 | Distributional shifts

Predictive analytics deals with the problem of inference; that is, analyzing patterns and making predictions from observational data (Ghahramani, 2015; Hastie et al., 2009). Formally, one infers an outcome $y \in Y$ based on an input $x \in X$ from a predictive model f (Jordan & Mitchell, 2015). As formalized in statistical learning theory, the performance of a predictive model depends on its ability to generalize well on out-of-sample observations. The core assumption is that both past and future observations are independently drawn from the same probability distribution over $X \times Y$. If this assumption is violated (i.e., $\mathbb{P}(X, Y) \neq \mathbb{P}(X', Y')$ between in-sample and out-of-sample observations), the performance of predictive models is likely to deteriorate for future observations. This is the case if a predictive model is deployed on data that stem from a different probability distribution than the training data. In the literature, this is referred to as *distributional shift* (or domain shift; Kouw & Loog, 2018).

Distributional shifts have been extensively studied for the specific requirements in computer vision and computational linguistics. These studies mostly draw upon common data

sets for benchmarking (e.g., handwritten digits) but do not involve operational data. Recently, there have been some technical contributions focusing on learning under distributional shifts (e.g., Ganin et al., 2016; Shen et al., 2018; Tzeng et al., 2017; Wang et al., 2019). These methods can be subsumed under the term "domain adaptation" or "domain adaptive learning." The objective of domain adaptive learning is to perform an end-to-end prediction task while simultaneously considering distributional shifts between two different domains (e.g., predicting product reviews for books based on product reviews written for movies). While domain adaptive learning has shown great potential with image and text data, its operational value in OM practice has not yet been studied.

There are approaches for handling distributional shifts but with a clearly different objective (Kouw & Loog, 2018; McNamara & Balcan, 2017; Pan & Yang, 2010). First, there is *model retraining* (Cui et al., 2018). Here, a model is (continuously) updated using data (x, y) with features and labels from the new operational setting, so that it adapts to the distributional shift in operational data. As in online learning, the latter requires continuous access to labels, which makes it suitable for make-to-stock manufacturing (e.g., fast-moving consumer goods). In contrast, such labels are not available for forthcoming orders in customized production because of which this approach is not applicable to our work. Second, there is *transfer learning* via fine-tuning (Kouw & Loog, 2018; McNamara & Balcan, 2017). Here, inferences are made between different predictive tasks (e.g., changing y from fault risk to cost) or between different populations. To this end, data from the new predictive task or new population are used to update the model weights. This is effective for operational settings with proxy data (Bastani, 2021). Yet, such data are typically unavailable for forthcoming orders in customized production. In sum, the above methods require labeled data from the deployment setting; therefore, none of these methods fulfill the objective of this work.

Distributional shifts represent a key hurdle for applying predictive analytics. Yet, despite its relevance, there is a scarcity of research on its practical implications (Simester et al., 2020). To the best of our knowledge, there is no research that suggests how to effectively mitigate distributional shifts in manufacturing. However, distributional shifts appear in all real-world operations and can lead to poor decision making. Motivated by the general trend in manufacturing toward increased product customization (Choi et al., 2021; Feng & Shanthikumar, 2018; Olsen & Tomlin, 2020), this paper addresses distributional shifts in operational data through the use of adversarial learning.

2.4 | Adversarial learning

The term "adversarial learning" refers to a general technique in predictive analytics whereby a neural network is supposed to learn two adversarial objectives (Goodfellow et al., 2014). For example, it allows one to train a neural network that has good prediction performance *and* where the representation of

the neural network simultaneously fulfills another constraint. Mathematically, the two adversarial objectives can be viewed as a two-player minimax game. Yet, implementing adversarial learning in practice is challenging. On the one hand, an appropriate optimization technique must be chosen to ensure convergence with state-of-the-art solvers (e.g., stochastic gradient descent), and, on the other hand, optimization of the two objectives must take place in the latent space of the neural network parameters.

Adversarial learning has first been introduced as part of generative adversarial networks (GANs; Goodfellow et al., 2014). GANs generate synthetic data (e.g., artificially created images) that are indistinguishable from samples drawn from a real data distribution (e.g., distribution of real images). In order to achieve this, two separate two neural networks are used: a generator G that generates a new synthetic sample and a discriminator D that estimates if a sample stems from the original distribution (i.e., sampled as in the training data) or from the model distribution (i.e., sampled from the generator). Both networks are then trained jointly with adversarial objectives: The discriminator is trained to distinguish the two distributions, while the generator model is trained to fool the discriminator. This results in a two-player minimax game with an optimal solution in which the generator distribution is equal to the data distribution.

Adversarial learning has also shown success in domain adaptive learning for image and text data (Ganin et al., 2016; Shen et al., 2018; Tzeng et al., 2017). Here, adversarial learning is applied to map features from different domains into a common latent space, so that the training procedure becomes an end-to-end prediction task. This is usually realized by training one neural network to achieve the best possible prediction performance, while a second adversarial network is trained to keep the feature distributions from both domains close. This idea is based on the theoretical findings of Ben-David et al. (2010, 2007), which suggest that, for good feature representations in cross-domain transfer, a discriminator should not be able to distinguish from which domain an observation originated.

In this paper, we tailor domain adversarial learning to OM decision making; that is, we address distributional shifts between different orders in customized production to improve job shop scheduling.

3 | EMPIRICAL SETUP AT AKER

3.1 | Job shop production at Aker

Our empirical application is carried out at Aker, headquartered in Lysaker, Norway. Aker is a leading engineering company in the energy sector, with an annual turnover of approximately USD 3.4 billion in 2021. The company covers the entire value chain, including fabrication engineering, purchasing, manufacturing, and delivery. We focus on a job shop production involving customer orders for large metal components that are supplied for the construction of oil
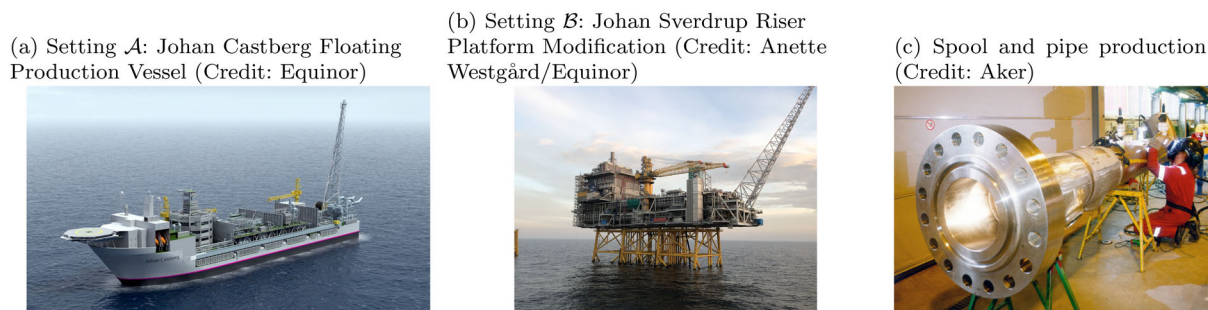
FIGURE 1 Empirical context at Aker. [Color figure can be viewed at wileyonlinelibrary.com]

platforms. Due to strong dependencies, delays in individual production orders can lead to substantial economic losses. Therefore, Aker puts great emphasis on order fulfillment in its component production sites.

In our numerical experiments, we use data from the two most recent order settings: "Johan Castberg Floating Production Vessel" (setting $\mathcal{A}$) and "Johan Sverdrup Riser Platform Modification" (setting $\mathcal{B}$). Setting $\mathcal{A}$ involves the production of topside modules for a floating production vessel. Setting $\mathcal{B}$ involves the production of components for the modification of an offshore oil platform. As can be observed in Figure 1, the two settings for which Aker supplies components differ radically. Both settings contain complex piping networks that consist of thousands of interconnected metal spools. Each of these spools corresponds to an individual production order that can have distinctive specifications in terms of material requirements, size, and shape. To avoid costly delays at the construction yard of the customer, it is crucial that the spools are available on time.

The order due dates of all spools are recorded in a centralized enterprise resource planning (ERP) system. The production management at Aker reviews the spools for which the engineering specifications and raw materials have arrived, that is, the backlog. This provides the basis for deciding which spools should be started next. Tardiness in the spool production (i.e., the actual finish date exceeds the order due date) can cause substantial disruptions in the value chain. In contrast, spools that finish too early result in excess inventory that must be managed in a limited outdoor space (recall, the spools are large, bulky, and heavy) and may lead to rework in the event of engineering change orders. For Aker, the cost of carrying excess inventory is substantial. By accurately estimating the throughput times of orders in the backlog, production managers can optimize their scheduling decisions to improve the on-time delivery performance (see Figure 2).

To meet order fulfillment targets, Aker follows a two-staged, predict-then-optimize approach: In the first stage, Aker estimates the throughput times for customer orders, and, in the second stage, production managers solve a job shop scheduling problem to reduce costs from early and tardy production orders (see Section 2.2 for an overview on job shop scheduling). However, because each order setting (i.e., construction of oil platform) served by Aker is unique, there is substantial heterogeneity between different customer

orders and thus distributional shifts in the operational data. Therefore, a naïve application of machine learning to predict throughput times may lead to a suboptimal prediction performance and eventually suboptimal scheduling decisions. This motivates our data-driven approach to address distributional shifts between order settings with adversarial learning.

3.2 | Empirical task

The task is to solve a job shop scheduling problem that optimizes decision making such that the cost of deviations from the order due dates is minimized. Specifically, we aim to schedule orders for setting $\mathcal{B}$ ("Johan Sverdrup Riser Platform Modification") while making use of historical data from setting $\mathcal{A}$ ("Johan Castberg Floating Production Vessel").

Aker provided us with operational data from the two order settings. The data for setting $\mathcal{A}$ comprise the production details of $n = 5830$ spools that were produced between January 2019 and September 2019. The data for setting $\mathcal{B}$ comprise the production details of $m = 3866$ spools that were produced between September 2019 and April 2020. Notably, there is no chronological overlap between the operational data from setting $\mathcal{A}$ and setting $\mathcal{B}$ (see Figure 3).

For job shop scheduling, we consider all information available to Aker at the time of scheduling the orders for the forthcoming setting $\mathcal{B}$; that is, we use historical data from setting $\mathcal{A}$ to schedule the orders belonging to setting $\mathcal{B}$. Hence, for both settings, we have access to spool-specific features. We denote these features by $\{(x_i^{\mathcal{A}})\}_{i=1}^n$ and $\{(x_i^{\mathcal{B}})\}_{i=1}^m$, respectively. For the historical setting $\mathcal{A}$, we additionally have information on the actual order throughput times that were observed in the past. We refer to them as "labels" and denote them by $\{(y_i^{\mathcal{A}})\}_{i=1}^n$. In contrast, for setting $\mathcal{B}$, we do not have such labels with information on order throughput times, because this is the forthcoming order setting for which we predict throughput times and schedule individual orders. In the following, we make use of the order throughput times for setting $\mathcal{B}$ but only for the purpose of evaluation.

The observed throughput times from setting $\mathcal{A}$ and setting $\mathcal{B}$ are shown in Figure 4, suggesting considerable differences. The average throughput time in setting $\mathcal{A}$ amounts to 32.0 days, whereas the average throughput time in setting $\mathcal{B}$ amounts to 35.2 days. A Welch's t-test confirms
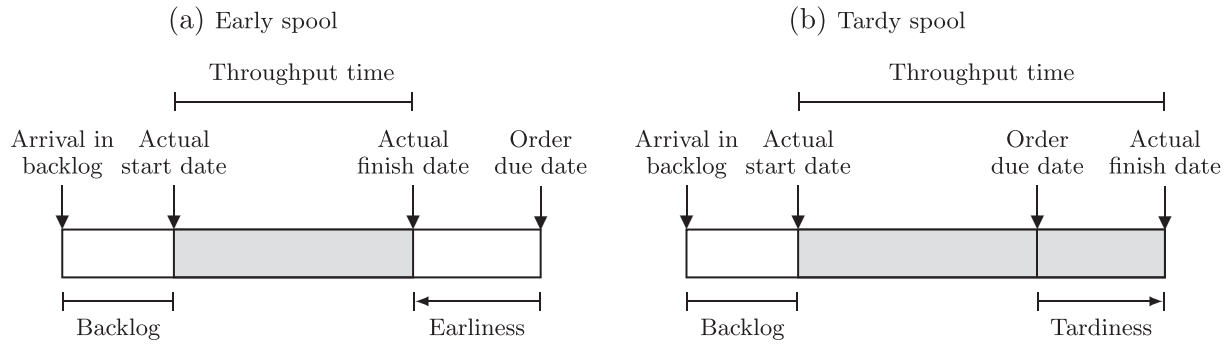
FIGURE 2   Example timelines for earliness and tardiness in spool production.
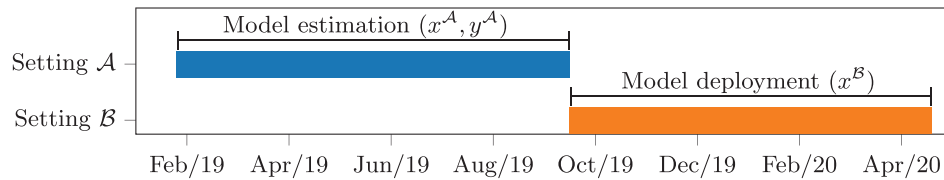

FIGURE 3   Spool production timelines of order settings $\mathcal{A}$ and $\mathcal{B}$. [Color figure can be viewed at wileyonlinelibrary.com]
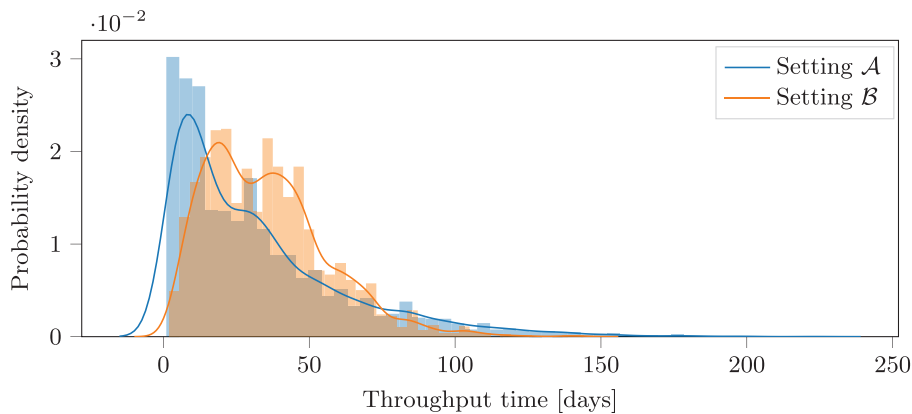

FIGURE 4   Distributions of throughput times for setting $\mathcal{A}$ and setting $\mathcal{B}$. [Color figure can be viewed at wileyonlinelibrary.com]

that the differences in throughput times are statistically significant ($p < 0.001$). Recall again that the throughput times from setting $\mathcal{B}$ are unknown at the time of prediction (i.e., when starting to produce spools for setting $\mathcal{B}$) and are only used for evaluation.

The data from both order settings contain $d = 20$ spool-specific features, which we use to predict the throughput times (Table 1). The features include detailed information about the spool-specific timelines, material specifications, and required processing steps. Note that Aker computes the features daily with the available information before and until the time point when the prediction is made. Because of that, a look-ahead bias is prevented.

All features are potentially subject to distributional shifts between the two order settings. For example, one setting may require spools with particularly fine tolerances that were not needed in previous settings. There may also be distributional shifts when the interrelations between features change. This can, for example, be the case if one order setting requires long and thin spools while the other requires short and thick spools.

3.3 | Exploratory analysis of distributional shifts

We now explore the distributional shifts between the spool-specific features of setting $\mathcal{A}$ and setting $\mathcal{B}$. Because our operational data comprise $d = 20$ features, we focus on the following multivariate methods for assessing the distributional shifts: (1) t-distributed stochastic neighbor embedding (t-SNE) and (2) adversarial validation. The results are summarized below.

First, we apply t-SNE (van der Maaten & Hinton, 2008) to investigate (dis-)similarities in the feature distributions. The t-SNE method is a nonlinear dimensionality reduction

TABLE 1 List of features.

| Feature	Description	Feature	Description
x_1	Material multiplication factor	x_{11}	Demolition multiplication factor
x_2	Insulation thickness around the spool	x_{12}	Min. design temperature of the fluid/gas in the spool
x_3	Length of the spool	x_{13}	Test pressure of the fluid/gas in the spool during testing
x_4	Dry weight of the spool	x_{14}	Operational pressure of the gas/fluid in the spool
x_5	Summed length of all welds of the spool	x_{15}	Max. operational temperature of the fluid/gas in the spool
x_6	Number of welds	x_{16}	Max. design temperature of the fluid/gas in the spool
x_7	Average weld diameter	x_{17}	Number of different materials needed to produce the spool
x_8	Average weld thickness	x_{18}	Number of different tasks to produce the spool
x_9	Planned length of the job	x_{19}	Max. average historical delay of the material deliveries for the spool
x_{10}	Planned sum of workline hours	x_{20}	Revision multiplication factor
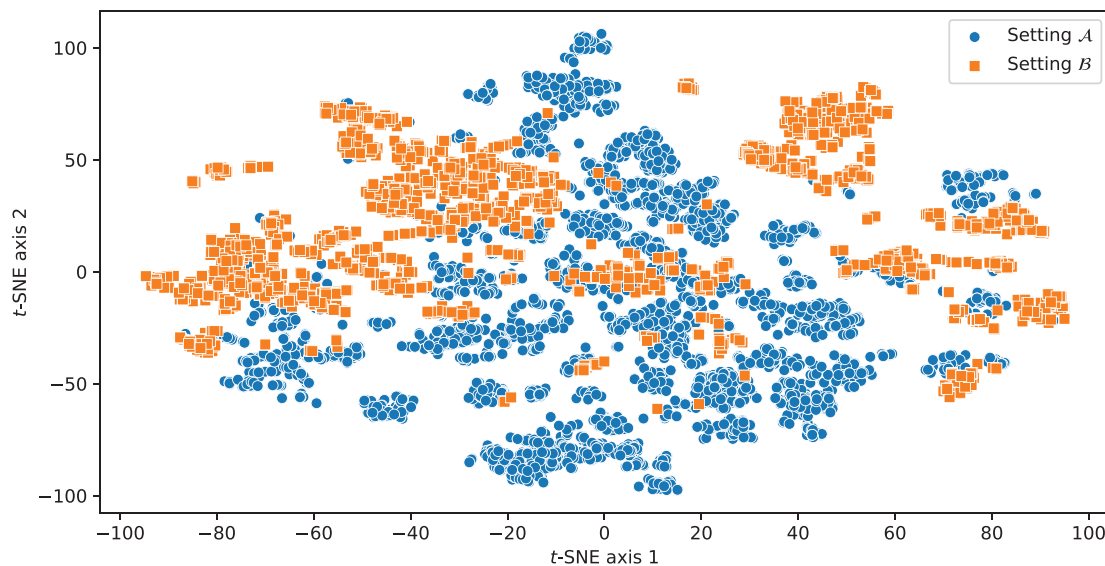
FIGURE 5 Two-dimensional representation of the spool-specific feature spaces based on *t*-distributed stochastic neighbor embedding (*t*-SNE). [Color figure can be viewed at wileyonlinelibrary.com]

technique that is specifically designed for visualizing high-dimensional data. The idea behind the *t*-SNE method is to convert the similarities between data points into joint probabilities and to minimize the Kullback–Leibler divergence between the joint probabilities of a low-dimensional embedding and the original feature space (van der Maaten & Hinton, 2008). This yields a low-dimensional representation that can be visualized.

We utilize *t*-SNE to assess whether the spool-specific features of setting $\mathcal{A}$ and setting $\mathcal{B}$ are distributed similarly. Figure 5 shows the feature representations for both settings in a two-dimensional space. Note that the axes do not have a specific meaning but only give an intuition about how the spool observations are distributed in the original feature space (i.e., $d = 20$). It can be observed that the feature representations form largely disjunct clusters with little overlap between the two settings. This provides strong evidence that the operational specifications of setting $\mathcal{A}$ and setting $\mathcal{B}$ are substantially different.

Second, we are interested in which features are particularly important in explaining the distributional shifts between both order settings. To achieve this, we draw upon adversarial validation (Pan et al., 2020). In adversarial validation, one trains a classifier that discriminates between features originating from setting $\mathcal{A}$ and setting $\mathcal{B}$. More formally, we learn a binary classifier to distinguish whether a feature x is drawn from $\mathcal{A}^X$ or $\mathcal{B}^X$. The labels are given by binary indicators that suggest the setting from which x was sampled. Provided there is no distributional shift between the two settings, a classifier should not be able to discriminate between features; that is, it should not perform better than a random guess. This would correspond to an area under the receiver operating characteristic curve (ROC-AUC) close or equal to 0.5. In the event of a distributional shift, the ROC-AUC would be significantly above 0.5. In this case, an analysis of feature importance can help to identify which features explain distributional shifts.

We implement adversarial validation via gradient boosting with decision trees (Ke et al., 2017). We run the analysis
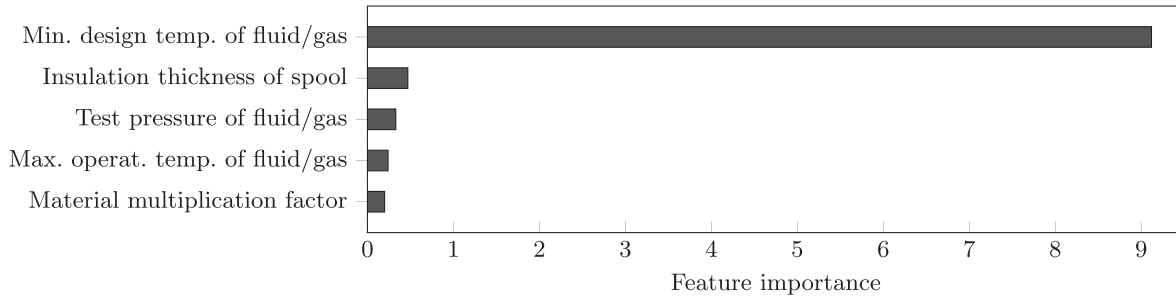
FIGURE 6 Features associated with the largest distributional shift.

over 100 different training and validation splits and consistently arrive at an out-of-sample ROC-AUC of 1.0. In other words, the classifier can perfectly discriminate between features originating from setting $\mathcal{A}$ and setting $\mathcal{B}$, thereby adding further evidence of distributional shifts. We then compute the average feature importance based on the mean absolute feature attribution (Lundberg et al., 2020). This allows us to identify the features that are the most important in explaining distributional shifts. The top five features associated with the largest distributional shift are listed in Figure 6. The results suggest that a large portion of the distributional shifts can be explained by differences in the production conditions and material specifications (i.e., minimum design temperature, required insulation thickness, test pressure, ...).

Overall, our exploratory analysis confirms that there are substantial distributional shifts between the two order settings. This can have a substantial effect on predictive models when training on data from setting $\mathcal{A}$ and predicting the operational outcomes (throughput times) for setting $\mathcal{B}$. As we show in the following, the straightforward use of predictive analytics can harm prediction performance and thus scheduling decisions.

4 | MODEL DEVELOPMENT[1]

In this section, we introduce our data-driven approach based on adversarial learning and job shop scheduling to support manufacturing operations in highly customized production. We first give a problem description (Section 4.1) and connect it with the concept of distributional shifts (Section 4.2). Then, we adapt adversarial learning to address the distributional shifts in our decision problem (Section 4.3).

4.1 | Problem description

We consider a job shop production, where production orders are scheduled such that the expected costs of earliness (i.e., completing the order too early) and tardiness (i.e., completing the order too late) are minimized. We assume that each order $i = 1, \ldots, m$ has a due date d_i and a stochastic throughput time

y_i. The actual throughput time y_i is unknown a priori and must thus be predicted from historical observations. We further assume that the to-be-completed production orders come from a forthcoming setting (called setting $\mathcal{B}$) and we have access to labeled data from a historical setting (called setting $\mathcal{A}$). Due to customized production, settings $\mathcal{A}$ and $\mathcal{B}$ are different. Formally, setting $\mathcal{A}$ involves n individual production orders, where every order i is described by order-specific features $x_i^{\mathcal{A}} \in \mathbb{R}^d$. Without loss of generality, $\mathcal{A}$ can also be a set of multiple historical order settings. Further, let each individual order be associated with an observed order throughput time $y_i^{\mathcal{A}} \in \mathbb{R}$. The planned and forthcoming setting $\mathcal{B}$ comprises m to-be-completed orders for which we would like to predict future throughput times. When starting to produce for order setting $\mathcal{B}$, we have access to order-specific features $x_i^{\mathcal{B}} \in \mathbb{R}^d$ but not to the throughput times $y_i^{\mathcal{B}} \in \mathbb{R}$ because they lie ahead of time.

To optimize the job shop production, manufacturers typically follow a predict-then-optimize approach: In the first stage, the throughput times for customer orders are predicted, and, in the second stage, a job shop scheduling problem is solved in order to optimize scheduling decisions. We formalize this in the following.

Stage 1. In the first stage, we estimate a predictive model $f : X \to Y$ to predict the throughput times $\hat{y}_i^{\mathcal{B}}$ for orders $i = 1, \ldots, m$ from the forthcoming setting $\mathcal{B}$. Here, we can make use of order-specific features $x_i^{\mathcal{B}}$ that characterize the individual production orders (e.g., material specifications, process configurations, timeline information). The predicted throughput times for setting $\mathcal{B}$ are given by $\hat{y}^{\mathcal{B}} = f(x^{\mathcal{B}})$.

The input for estimating the predictive model f is as follows. For the historical setting $\mathcal{A}$, we have access to order-specific features $\{(x_i^{\mathcal{A}})\}_{i=1}^n$ and the ground-truth throughput times $\{(y_i^{\mathcal{A}})\}_{i=1}^n$. For the forthcoming setting $\mathcal{B}$, we only have access to the order-specific features $\{(x_i^{\mathcal{B}})\}_{i=1}^m$ but not the throughput times as these lie ahead of time. That is, we have a labeled data set sampled from setting $\mathcal{A}$ and an unlabeled data set sampled from the marginal distribution of setting $\mathcal{B}$ over X. Hence, the input is given by

$$\left\{ \left(x_i^{\mathcal{A}}, y_i^{\mathcal{A}} \right) \right\}_{i=1}^n \sim_{\text{iid}} (\mathcal{A})^n \quad \text{and} \quad \left\{ \left(x_i^{\mathcal{B}} \right) \right\}_{i=1}^m \sim_{\text{iid}} (\mathcal{B}^X)^m.$$

$$(2)$$

To estimate f, we aim at minimizing the expected error ϵ_B between the predicted throughput times and observed throughput times for setting B. This yields the following objective:

$$\min_f \epsilon_B(f) = \min_f \mathbb{E}_{(x^B, y^B) \sim B} \left[\mathcal{L}(y^B, \hat{y}^B) \right]$$

$$\text{given data } \left\{ \left(x_i^A, y_i^A \right) \right\}_{i=1}^n \text{ and } \left\{ \left(x_i^B \right) \right\}_{i=1}^m,$$

$$(3)$$

where $\mathcal{L}$ denotes a convex loss function.[2] Here, Equation (3) is important: We aim at minimizing the expected error for setting B while having access to order-specific features for both setting A and setting B but only observed outcomes for setting A. We later adapt adversarial learning to make such predictions of throughput times while accounting for the heterogeneity between the order-specific features of setting A and setting B (see Section 4.3). In contrast to that, off-the-shelf machine learning would only minimize the expected error for setting A without considering the distributional shift between the order-specific features of setting A and setting B.

Stage 2. In the second stage, the scheduling task for the forthcoming setting B is formalized as an integer linear programming problem that determines a production sequence of the m to-be-completed production orders given T available time slots. Formally, let $\hat{y}_i$ denote the estimated order throughput time and d_i the due date of a given production order i. The per time unit costs of earliness and tardiness are given by $c^{(early)}, c^{(tardy)} \in \mathbb{R}^{\geq 0}$, respectively. Further, let K_t define the total number of production orders (i.e., capacity) that can be processed at time t. Then, the optimal production sequence can be solved via

$$\min_{z_{it} \in \{0,1\}} \sum_{t=1}^T \sum_{i=1}^m z_{it} \left[c^{(early)} \max\{0, (d_i - t) - \hat{y}_i\} \right.$$

$$\left. + c^{(tardy)} \max\{0, \hat{y}_i - (d_i - t)\} \right] \quad (4a)$$

$$\text{s.t.} \quad \psi(z, \hat{y}, t) \leq K_t, \qquad \text{for } t = 1, \ldots, T, \quad (4b)$$

$$\sum_{t=1}^T z_{it} = 1, \quad \text{for } i = 1, \ldots, m, \quad (4c)$$

where the binary decision variable z_{it} determines whether production order i should be started at time t and where the function $\psi(z, \hat{y}, t)$ counts the number of production orders that are produced in parallel. Here, the first constraint ensures that the available capacity is not exceeded, while the second constraint ensures that all production orders are fulfilled. We solve the optimization problem using the branch-and-cut implementation for mixed-integer problems from the GNU Linear Programming Kit (GLPK). Importantly, the scheduling task makes use of the throughput times $\hat{y}_i$, which are not given ex ante but—analogous to OM practice—must be predicted a priori before scheduling.

The above problem is formulated as a predict-then-optimize approach due to three important practical benefits: (1) It follows the current practice in order fulfillment. For example, a predict-then-optimize approach is consistent with decision making at our case company Aker and other manufacturing firms. (2) It offers great flexibility with regard to the chosen machine learning model. In particular, it allows manufacturers to use existing machine learning tools from their company. (3) It allows manufacturers to incorporate expert knowledge. For example, manufacturers can assess the accuracy of the predictions before proceeding to the scheduling stage.

Crucial to the above approach are accurate predictions of throughput times in the first stage. The reason is that incorrect predictions will lead to suboptimal production schedules and therefore additional costs. This can be formally seen in Equation (4a), where inaccurate predictions of throughput times $\hat{y}_i$ negatively affect the overall production schedule. Hence, by accurately predicting throughput times (*stage* 1), manufacturers can find optimal scheduling sequences (*stage* 2), such that the cumulative costs of earliness and tardiness are minimized. However, predicting throughput times is particularly challenging when manufacturers produce highly customized products with nonstandard specifications (Cohen et al., 2003). Such customized products are characterized by large between-order heterogeneity and thus data samples that are not identically distributed, which, in turn, violates a standard assumption of machine learning (cf. Hastie et al., 2009). Hence, off-the-shelf machine learning models (e.g., standard deep neural networks) may give poor predictions of throughput times. The reason is that different specifications in customized production introduce distributional shifts between different customer order settings. This motivates an approach that accurately predicts order throughput times while accounting for distributional shifts between different order settings.

4.2 | Distributional shifts between order settings

We now connect the heterogeneity among order settings to the concept of distributional shifts and thereby motivate the use of adversarial learning to give better predictions in stage 1. Recall that we consider different order settings where we denote the historical order setting by A (in our empirical context, 5830 component orders for a floating production vessel) and the forthcoming order setting by B (in our empirical context, 3866 component orders for an offshore oil platform). Here, we specifically focus on manufacturing settings with high degrees of product customization. In this particular context, forthcoming orders from new customers can involve entirely new specifications for which there are no historical observations. Varying specifications typically lead to between-order heterogeneity. Formally, the between-order heterogeneity is expressed by a distribution over $X \times Y$ that changes between model estimation (setting A) and model

deployment (setting $\mathcal{B}$). This is stated in the following definition of *a distributional shift*.

Definition 4.1 (Distributional shift). A distributional shift is a change in the joint probability distribution between the data from setting $\mathcal{A}$ that is used for model estimation and the data from setting $\mathcal{B}$ that is used during model deployment, that is,

$$\mathbb{P}_{\mathcal{A}}(X, Y) \neq \mathbb{P}_{\mathcal{B}}(X, Y). \tag{5}$$

Building upon the concept of distributional shifts, we now explain why a naïve application of off-the-shelf machine learning does not solve our task, and thereby we motivate the use of adversarial learning. In particular, we consider a specific case of distributional shift where the difference in the joint distribution $\mathbb{P}(X, Y)$ between setting $\mathcal{A}$ and setting $\mathcal{B}$ results from a difference in the marginal distribution of X (i.e., $\mathbb{P}(X)$), whereas the conditional distribution of Y given X (i.e., $\mathbb{P}(Y \mid X)$) remains unchanged between the two settings. Formally, we address a distributional shift of the type

$$\mathbb{P}_{\mathcal{A}}(X, Y) = \mathbb{P}(Y \mid X)\mathbb{P}_{\mathcal{A}}(X) \neq \mathbb{P}(Y \mid X)\mathbb{P}_{\mathcal{B}}(X) = \mathbb{P}_{\mathcal{B}}(X, Y),$$

$$\tag{6}$$

where $\mathbb{P}_{\mathcal{A}}(Y \mid X) = \mathbb{P}_{\mathcal{B}}(Y \mid X) = \mathbb{P}(Y \mid X)$, but $\mathbb{P}_{\mathcal{A}}(X) \neq \mathbb{P}_{\mathcal{B}}(X)$. This form of distributional shift is known in the literature as covariate shift (Kouw & Loog, 2018). In fact, distributional shifts in the form of covariate shifts are common at Aker and across OM practice. The latter essentially states that the specifications change between orders (i.e., $\mathbb{P}_{\mathcal{A}}(X) \neq \mathbb{P}_{\mathcal{B}}(X)$). For example, at Aker, one setting may require thin and long spools while the other may require thick and short spools. The former essentially states that the process behind manufacturing products is comparable; that is, orders with identical specifications have the same throughput times regardless of whether they belong to setting $\mathcal{A}$ or $\mathcal{B}$ (i.e., $\mathbb{P}_{\mathcal{A}}(Y \mid X) = \mathbb{P}_{\mathcal{B}}(Y \mid X)$). For example, at Aker, thin (and long) spools will take the same time for production independent of whether the thin (and long) spool is later used in a floating production vessel or an offshore oil platform.

In a naïve application of machine learning, one would simply estimate f only based on $(x^{\mathcal{A}}, y^{\mathcal{A}})$. This has two key disadvantages (which later present two salient differences to our proposed approach). First, predictive models from off-the-shelf machine learning ignore the operational data from the forthcoming order, that is, $x^{\mathcal{B}}$. However, such operational data characterizing forthcoming production orders are already available at the time of scheduling and could be used to improve the predictions and therefore the scheduling decisions. Second, predictive models from off-the-shelf machine learning optimize against $\min_f \mathbb{E}_{(x,y)\sim\mathcal{A}}[\mathcal{L}(y, \hat{y})]$ and not $\min_f \mathbb{E}_{(x,y)\sim\mathcal{B}}[\mathcal{L}(y, \hat{y})]$. That is, off-the-shelf machine learning optimizes the prediction performance for data coming from the historical probability distribution of data from setting $\mathcal{A}$ and not that of the forthcoming setting $\mathcal{B}$. However, under a distributional shift $\mathbb{P}_{\mathcal{A}}(X, Y) \neq \mathbb{P}_{\mathcal{B}}(X, Y)$,

both probability distributions are different; therefore, the optimization will not solve our objective from Equation (3). The reason is that off-the-shelf machine learning makes the assumption of i.i.d. sampling (cf. Hastie et al., 2009) and thus that the distribution over $X \times Y$ remains unchanged between model estimation (setting $\mathcal{A}$) and model deployment (setting $\mathcal{B}$). This assumption does not hold in our manufacturing setting with high degrees of product customization. As a result, the performance of such off-the-shelf predictive models will deteriorate when deployed to a forthcoming setting $\mathcal{B}$ and will lead to suboptimal scheduling decisions.

4.3 | Proposed adversarial learning approach for predicting throughput times

4.3.1 | Overview

In the following, we address the objective from Equation (3) through the use of adversarial learning. For this, we integrate the observed data $(x^{\mathcal{A}}, y^{\mathcal{A}})$ from setting $\mathcal{A}$ and the order-specific features $x^{\mathcal{B}}$ from setting $\mathcal{B}$ into the estimation of the predictive model f. This is referred to as an unsupervised domain adaptation problem. To provide a solution approach, we adapt adversarial learning to account for the different distributions behind $\mathcal{A}$ and $\mathcal{B}$. Specifically, we take advantage of Wasserstein distance guided representation learning (WDGRL; Shen et al., 2018). WDGRL has been previously used for classification tasks in computer vision and computational linguistics but not for OM decisions. Using adversarial learning, we predict the throughput times while addressing the distributional shift and then solve the scheduling problem in Equation (4a)–(4c) via integer optimization. Later, we confirm the effectiveness of adversarial learning over off-the-shelf machine learning for making job shop scheduling decisions under distributional shifts.

Formally, our aim is to predict throughput times under distributional shifts, that is, to achieve a low expected error in the forthcoming setting $\mathcal{B}$, so that the scheduling decisions can be optimized. Because we have no observed outcomes for setting $\mathcal{B}$, we cannot directly optimize the objective in Equation (3). Nevertheless, the expected error in unsupervised domain adaptation problems can be bounded as stated in Remark 4.1 (adapted from Ben-David et al., 2010, 2007; Kouw & Loog, 2018; Redko et al., 2017). To state the remark, we first need a definition of the Wasserstein distance.

Definition 4.2 (Wasserstein distance). The Wasserstein-1 (or Earth-Mover) distance between two probability distributions $\mathcal{A}$ and $\mathcal{B}$ is defined as,

$$W(\mathcal{A}, \mathcal{B}) = \inf_{\gamma \in \Pi(\mathcal{A}, \mathcal{B})} \mathbb{E}_{(v,w)\sim\gamma}[\|v - w\|], \tag{7}$$

where $\Pi(\mathcal{A}, \mathcal{B})$ is the set of all joint distributions $\gamma(v, w)$ with marginals $\mathcal{A}$ and $\mathcal{B}$.

Intuitively, the Wasserstein distance denotes the minimal amount of probability mass that must be transported (e.g., minimum expected transportation cost) from one distribution to the other to make them identical (Arjovsky et al., 2017).

Remark 4.1 Redko et al. (2017); Lemma 1. The prediction error for setting $\mathcal{B}$ (i.e., $\epsilon_{\mathcal{B}}$) can be bound by the sum of the prediction error for setting $\mathcal{A}$ (i.e., $\epsilon_{\mathcal{A}}$) and the Wasserstein distance $W(\mathcal{A}, \mathcal{B})$ between the feature distributions of settings $\mathcal{A}$ and $\mathcal{B}$, that is,

$$\epsilon_{\mathcal{B}} \le \epsilon_{\mathcal{A}} + W(\mathcal{A}, \mathcal{B}), \tag{8}$$

under some technical assumptions; see Redko et al. (2017).

The above remark assumes a machine learning classifier and provides the following theoretical motivation for our learning approach (see Supporting Information A for a more detailed discussion of the error bound and the underlying technical assumptions). First, the upper bound of the prediction error depends on how well we can make predictions for setting $\mathcal{A}$. Second, the upper bound of the prediction error should increase when the probability distributions of both settings drift apart. This motivates our adversarial learning approach where we aim to make inferences under two adversarial objectives: (1) Our first objective is to estimate a function with a low prediction error on setting $\mathcal{A}$. This is achieved by minimizing the loss between the actually observed outcomes and predictions from setting $\mathcal{A}$. (2) Our second objective accounts for the distance term $W(\mathcal{A}, \mathcal{B})$, whereby we learn latent feature representations of the order-specific features from both settings that are close to each other. More formally, we minimize the Wasserstein distance between the two feature distributions of setting $\mathcal{A}$ and setting $\mathcal{B}$. As such, we aim for a good prediction performance in the known setting $\mathcal{A}$, but draw upon a representation that also generalizes well to operational data from the forthcoming setting $\mathcal{B}$. The two aforementioned objectives are adversarial to each other (e.g., a close feature distribution does not imply a low error on setting $\mathcal{A}$ and vice versa). In the following, we formalize both objectives in a minimax game.

4.3.2 | Model specification

Our adversarial learning approach is composed of three functions as follows (Figure 7). (1) A shared feature extractor $f_e : \mathbb{R}^d \to \mathbb{R}^l$ maps order-specific features from the d-dimensional input space X of both order settings into a common latent space $\mathbb{R}^l$. This allows our approach to learn a shared representation of the latent feature distributions from both the historical and the forthcoming order setting. (2) A regressor $f_r : \mathbb{R}^l \to \mathbb{R}$ outputs the prediction, that is, the throughput time given the latent features. (3) A so-called critic $f_c : \mathbb{R}^l \to \mathbb{R}$ is used to estimate the Wasserstein distance between the latent feature distributions. We implement f_e, f_r, and f_c as parameterized differentiable functions given by

fully connected linear feed-forward neural networks. Upon deployment, predictions are then made using $f_r \circ f_e$. That is, for input x_i, we compute the predicted throughput time via $\hat{y}_i = f_r(f_e(x_i))$.

The functions f_e, f_r, and f_c are used in the two adversarial objectives as follows. The first adversarial objective ($\mathcal{L}_{\text{reg}}$) is to minimize expected prediction error for setting $\mathcal{A}$ and thus to learn predictions of the throughput time using data from the historical setting $\mathcal{A}$. It involves $f_r \circ f_e$, which outputs the predictions. Formally, we can calculate the predicted outcome $\hat{y}_i$ via $f_r(f_e(x_i))$ for any x_i sampled from setting $\mathcal{A}$. The second adversarial objective ($\mathcal{L}_{\text{was}}$) aims to minimize the Wasserstein distance between settings $\mathcal{A}$ and $\mathcal{B}$. Hence, it is based on $f_c \circ f_e$, so that the distance between the latent feature distributions of the historical and forthcoming setting is minimized.

Adversarial objective 1 ($\mathcal{L}_{\text{reg}}$): Minimizing the expected prediction error for setting $\mathcal{A}$

The first objective is to achieve a low expected prediction error for setting $\mathcal{A}$. For this, we compute the prediction error for labeled samples from setting $\mathcal{A}$. The loss is given by

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{(x,y) \sim \mathcal{A}} \ \mathcal{L}(f_r(f_e(x)), y). \tag{9}$$

Adversarial objective 2 ($\mathcal{L}_{\text{was}}$): Minimizing the Wasserstein distance between settings $\mathcal{A}$ and $\mathcal{B}$

The second objective is to minimize the Wasserstein distance between settings $\mathcal{A}$ and $\mathcal{B}$ in the latent feature space. Recall that, intuitively, the Wasserstein distance denotes the minimal amount of probability mass that must be transported (e.g., minimum expected transportation cost) from one distribution to the other to make them identical (Arjovsky et al., 2017). That is, we yield

$$W(\mathcal{A}, \mathcal{B}) = \inf_{\gamma \in \Pi(\mathcal{A}, \mathcal{B})} \mathbb{E}_{(v,w) \sim \gamma}[\|v - w\|], \tag{10}$$

where $\Pi(\mathcal{A}, \mathcal{B})$ is the set of all joint distributions $\gamma(v, w)$ with marginals $\mathcal{A}$ and $\mathcal{B}$.

However, computing $W(\mathcal{A}, \mathcal{B})$ directly is computationally infeasible. As a remedy, we make use of the critic function $f_c : \mathbb{R}^l \to \mathbb{R}$ (Arjovsky et al., 2017) by rewriting Equation (10) using the Kantorovich–Rubinstein duality, that is,

$$\sup_{\|f_c\|_L \le 1} \mathbb{E}_{x \sim \mathcal{A}^X} [f_c(f_e(x))] - \mathbb{E}_{x \sim \mathcal{B}^X} [f_c(f_e(x))], \tag{11}$$

where f_c must be 1-Lipschitz. This is achieved by adding a gradient penalty loss $\mathcal{L}_{\text{grad}}$ (Gulrajani et al., 2017), which penalizes the norm of the gradient itself. For this, we sample points $\hat{x}$ uniformly along straight lines between points sampled from $\mathcal{A}$ and $\mathcal{B}$. The gradient penalty term can then be written as

$$\mathcal{L}_{\text{grad}} = \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} f_c(\hat{x})\|_2 - 1)^2 \right]. \tag{12}$$

Further details on the gradient penalty loss are provided in Supporting Information B.
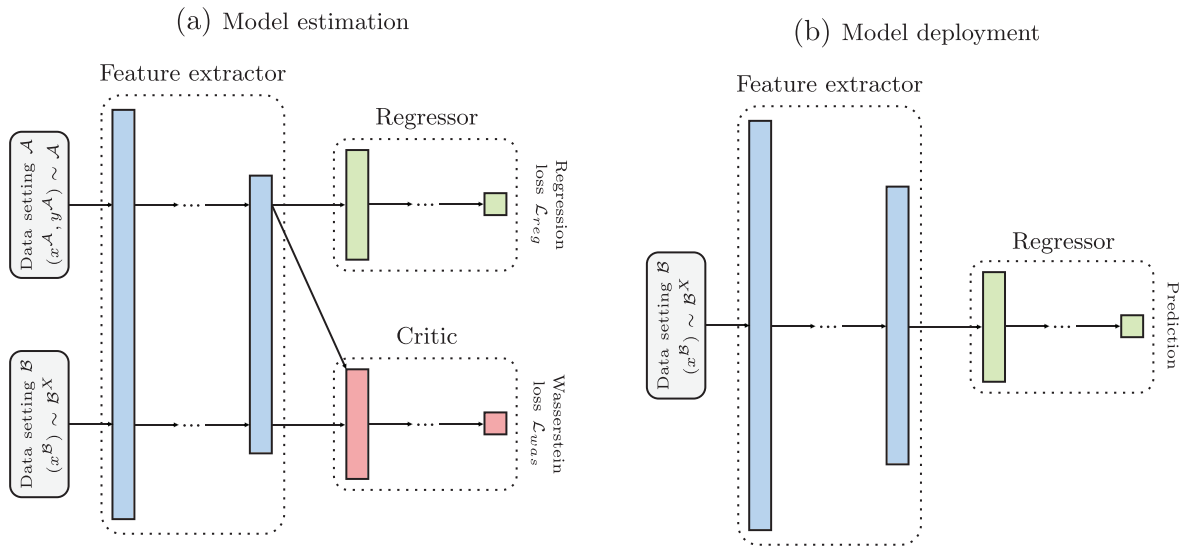
FIGURE 7   Model specification based on the feature extractor f_e, the regressor f_r, and the critic f_c. The use of the model depends on whether (a) parameters should be estimated or (b) predictions should be made upon deployment. [Color figure can be viewed at wileyonlinelibrary.com]

Altogether, the loss to compute the Wasserstein distance is then given by

$$\mathcal{L}_{\text{was}} = \mathbb{E}_{x \sim A^X}\big[f_c(f_e(x))\big] - \mathbb{E}_{x \sim B^X}\big[f_c(f_e(x))\big] - \beta\,\mathcal{L}_{\text{grad}}, \quad (13)$$

where β is a gradient penalty weight. By maximizing the Wasserstein loss $\mathcal{L}_{\text{was}}$, we find the supremum in Equation (11) that estimates the Wasserstein distance $W(\mathcal{A}, \mathcal{B})$ between the probability distributions of both settings.

4.3.3 | Estimation procedure

We now combine the two adversarial objectives given by the loss functions $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{was}}$ into a joint learning objective. Formally, this joint objective is given by the following minimax game:

$$\min_{f_e, f_r}\{\mathcal{L}_{\text{reg}} + \alpha \max_{f_c} \mathcal{L}_{\text{was}}\}, \quad (14)$$

where α is a constant that weights the Wasserstein loss. Equation (14) aims at reducing the expected prediction error on the historical setting $\mathcal{A}$ through $\mathcal{L}_{\text{reg}}$, while simultaneously finding the supremum in Equation (11) by maximizing $\mathcal{L}_{\text{was}}$ over f_c. The latter allows us to estimate the Wasserstein distance between the feature distributions and minimize it along with $\mathcal{L}_{\text{reg}}$ over f_e and f_r. Note that the gradient penalty is used during maximization in order to estimate the Wasserstein distance, but the Wasserstein distance estimate itself does not contain the gradient penalty term (as seen in Equation 11). Hence, the gradient penalty is not used during minimization; that is, for the min operation in Equation (14), we use the Wasserstein loss without the gradient penalty term.

In our implementation, we optimize the overall objective by alternating gradient descent following Shen et al. (2018). In every step, we first train the critic function to close opti-

mality (according to the max operation in Equation 14), and then update the feature extractor and regressor by minimizing the regression loss, as well as the Wasserstein distance (i.e., the Wasserstein loss without gradient penalty) estimated by the critic. We further set the weights in the loss function, that is, α from Equation (14) and β from Equation (13) to a default value of 1, so that we give equal weight to each part of the corresponding loss. We found that, regardless of the choice, the overall performance remains robust (see Supporting Information E).

Upon model deployment, we only need the feature extractor and regressor (see Figure 7) to make predictions. For order-specific features $x_i^{\mathcal{B}}$ from order setting $\mathcal{B}$, we predict the throughput time via

$$\hat{y}_i^{\mathcal{B}} = f_r\big(f_e\big(x_i^{\mathcal{B}}\big)\big). \quad (15)$$

Informed by Remark 4.1, this should then also minimize the prediction error $\epsilon_{\mathcal{B}}$ for the forthcoming order setting, and therefore address Equation (3), that is, the objective of our prediction task in stage 1. Hence, our adversarial learning approach should achieve superior predictions in setting $\mathcal{B}$ compared to off-the-shelf machine learning methods that only focus on minimizing $\epsilon_{\mathcal{A}}$ (while ignoring operational data from setting $\mathcal{B}$). Afterward, we use the predictions to compute the corresponding optimal scheduling sequence by solving the integer optimization in Equation (4a)–(4c), that is, the objective of our scheduling task in stage 2.

4.4 | Connecting prediction performance and scheduling costs

We now provide arguments for why our proposed adversarial learning approach is effective for solving our decision problem. Previously, in Section 4.2 and Section 4.3, we

motivated the use of adversarial learning to achieve better predictions compared to off-the-shelf machine learning in the presence of distributional shifts. Here, we discuss how the prediction performance translates to scheduling decisions and affects the scheduling costs. We first observe that the global minimum of scheduling cost is achieved for the "oracle" prediction in our decision problem, that is, when the prediction error $\epsilon_\mathcal{B}$ is zero. Then, we use this argument to support the use of our adversarial learning approach that minimizes $\epsilon_\mathcal{B}$ over off-the-shelf machine learning, which focuses only on minimizing $\epsilon_\mathcal{A}$.

When we discuss the scheduling costs in the following, we refer to the realized costs of earliness and tardiness (see Section 2.2) that result from a scheduling decision after the production is finished. For clarity, we provide an explicit definition below.

Definition 4.3. The realized scheduling cost for a job shop scheduling problem is defined as

$$\bar{C}(\mathbf{z}) = \sum_{t=1}^{T} \sum_{i=1}^{m} z_{it} \, [c^{(early)} \max \{0, (d_i - t) - y_i\}$$
$$+ c^{(tardy)} \max \{0, y_i - (d_i - t)\}], \qquad (16)$$

where m is the number of production orders, T is the number of available time slots, $z_{it} \in \{0, 1\}$ are binary decision variables that determine at which time t the production order i started (i.e., $\sum_{t=1}^{T} z_{it} = 1$ for $i = 1, \dots, m$), $c^{(early)}$ and $c^{(tardy)} \in \mathbb{R}^{\geq 0}$ are per time unit costs of earliness and tardiness, d_i is a given due date of order i, and y_i is the observed/realized throughput time of order i.

The realized scheduling cost depends on a scheduling decision $\mathbf{z} = (z_{11}, \dots, z_{1T}, z_{21}, \dots, z_{mT})$, which was made at time $t = 0$, that is, *prior* to the beginning of the production process, whereas the realized scheduling cost is measured *after* the production is finished and actual throughput times are observed. In stage 2 of our approach, the scheduling decision $\mathbf{z}$ is calculated by solving the optimization problem in Equation (4a)–(4c) given predictions for throughput times $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_m)$. Note that the realized scheduling cost function is of the same form as the cost function that we minimize in stage 2 of our approach (see Equation 4a). In fact, for the oracle predictions ($\hat{\mathbf{y}}_{\text{oracle}}$) that are equal to the true realized $\mathbf{y}$ (i.e., when $\hat{\mathbf{y}} = \hat{\mathbf{y}}_{\text{oracle}} = \mathbf{y}$), the two cost functions are equivalent. Since a scheduling decision $\mathbf{z}$ is an argument that minimizes the cost function in Equation (4a), a decision that is calculated based on the oracle predictions $\hat{\mathbf{y}}_{\text{oracle}}$ is thereby a decision that minimizes the realized scheduling costs. In other words, the global minimum of the realized scheduling cost is achieved for a decision that is calculated based on the oracle prediction. We formalize this in Lemma C.1 in Supporting Information C.

Therefore, perfectly accurate predictions of throughput times (with error $\epsilon_\mathcal{B}$ equal to zero) allow for optimal schedul-

ing decisions that minimize the realized scheduling costs. However, the oracle prediction is a theoretical construct that is generally not available in OM practice. Rather, we have prediction algorithms with varying errors in prediction. Intuitively, the result in Lemma C.1 in the Supporting Information suggests that, as we diverge from the "oracle" prediction and increase the prediction error (i.e., as we increase the MAE), the realized scheduling costs will increase correspondingly. In other words, the less accurate the predictions are (i.e., the larger the prediction MAE is), the larger the resulting scheduling costs will be. Hence, this underpins why our adversarial learning approach leads to superior decision making compared to off-the-shelf machine learning: Our approach aims to minimize $\epsilon_\mathcal{B}$, and therefore optimizes against the oracle predictions ($\hat{\mathbf{y}}_{\text{oracle}}$). In contrast, off-the-shelf machine learning minimizes $\epsilon_\mathcal{A}$, and therefore does *not* optimize against the oracle predictions under a distributional shift. To sum up, our approach directly optimizes against the objective in Equation (3) and thus achieves lower $\epsilon_\mathcal{B}$, which then translates into better scheduling decisions and ultimately cost savings.

5 | NUMERICAL EXPERIMENTS

In the following, we conduct a series of numerical experiments based on job shop scheduling to evaluate how distributional shifts in customized production affect production schedules, and therefore to study the operational value of our proposed adversarial learning approach. Further, to better understand the underlying mechanism of our approach, we further vary the operational setup across the following dimensions: (1) the magnitude of the distributional shift, (2) varying production line capacities, (3) varying cost parameters, (4) different distributions of the error term, and (5) different nonlinearities in the operational data.

5.1 | Experimental setup

In the following, we set up a simulation where we vary the magnitude of distributional shifts. The simulation is designed to mimic decision making in practice where the throughput times must be predicted before solving the job shop scheduling problem.

We simulate data for the features $x_i^\mathcal{A}$ and $x_i^\mathcal{B}$ as follows. We first use the actual features from Aker from settings $\mathcal{A}$ and $\mathcal{B}$ to estimate the corresponding means, $\mu_\mathcal{A}$ and $\mu_\mathcal{B}$, and covariance matrices $\Sigma_\mathcal{A}$ and $\Sigma_\mathcal{B}$, respectively. For the historical setting $\mathcal{A}$, we then sample features from a multivariate Gaussian distribution with mean $\mu_\mathcal{A}$ and covariance matrix $\Sigma_\mathcal{A}$. This gives the samples $\{(x_i^\mathcal{A})\}_{i=1}^{n}$ with $n = 5830$ analogous to the dimension of the historical order setting $\mathcal{A}$ at Aker. For the forthcoming setting $\mathcal{B}$, we set up the sampling such that we can vary the magnitude of the distributional shift by introducing a mean shift in the direction of the difference between the means $\mu_\mathcal{B}$ and $\mu_\mathcal{A}$ (i.e., a mean shift from

setting $\mathcal{A}$ toward setting $\mathcal{B}$). For this, we define the difference vector $v_{\text{diff}} = \mu_{\mathcal{B}} - \mu_{\mathcal{A}}$. Then, we sample features from a multivariate Gaussian distribution with mean $\mu_{\mathcal{A}} + \theta \cdot v_{\text{diff}}$ and covariance matrix $\Sigma_{\mathcal{B}}$, where parameter θ is used to control for the magnitude of the distributional shift (i.e., larger values for θ introduce larger distributional shifts). For a given θ, this gives the samples $\{(x_i^{\mathcal{B}})\}_{i=1}^m$ with $m = 3866$ analogous to the dimensions of the forthcoming order setting $\mathcal{B}$ at Aker.

To simulate throughput times, we need a data-generating function $\phi : X \to Y$, so that we can generate throughput times conditional on some given features (this is needed since we have different features depending on the magnitude of the distributional shift in our simulation). We thus follow standard practice in machine learning (e.g., Shalit et al., 2017; Yoon et al., 2018) where so-called semisynthetic data sets are used for modeling outcomes (in our case: throughput times). Specifically, we use predictive modeling to capture the data-generating process ϕ for Aker data in order to mimic the real-world setting at Aker. Here, the underlying choice of the machine learning model is crucial, because choosing some models may result in unfair advantages that bias later comparisons. For example, choosing a linear model for ϕ would strongly favor a linear model during evaluation. Similarly, choosing a neural network would favor our method because the structure of nonlinearities would be modeled in a similar way. Therefore, we use a nonlinear tree-based method, that is, a random forest. By using structurally different nonlinearity to generate the data, we ensure that none of the methods has an unfair advantage later.[3] Formally, the throughput times are simulated by using the features via $y = \phi(x) + \eta$, where ϕ is estimated using Aker data and where η is Gaussian noise, that is, $\eta \sim N(0, \sigma_y)$, with σ_y being the standard deviation estimate of y. For a given θ, we thus obtain the simulated throughput times $\{(y_i^{\mathcal{A}})\}_{i=1}^n$, and $\{(y_i^{\mathcal{B}})\}_{i=1}^m$, respectively.

We use the following operational setup: (1) The magnitude of the distributional shift is controlled by the parameter θ. In all of our numerical experiments, we report results for $\theta = 1, 2, 3, 4$ in order to examine how scheduling decisions are affected by different distributional shifts. (2) The production line capacity in our main numerical experiment is set to $K_t = 70$. We later also report results for $K_t = 50$ and thereby account for settings with smaller capacities such that production orders have to compete for production lines. (3) We set the costs to $c^{\text{early}} = c^{\text{tardy}} = 1$ so that both earliness and tardiness are equally costly. We later also account for settings where overdue deliveries are more costly than finishing early (i.e., $c^{\text{tardy}} = 2$). (4) We study how the distribution of the error term η in our simulation affects scheduling costs. We thus change the Gaussian distribution to a uniform distribution with short tails. Formally, we use $\eta \sim \text{Unif}(-\frac{\sqrt{12}\hat{\sigma}_y}{2}, \frac{\sqrt{12}\hat{\sigma}_y}{2})$ where the choice for the minimum and the maximum value is designed such that the standard deviation of η remains equal to the standard deviation from before. (5) We finally explore whether our results remain robust for varying nonlinearities in the operational data by changing the form of the data-generating process that we use to simulate the through-

put times. Here, we repeat our numerical experiments where ϕ is given by gradient boosting with decision trees.

Throughout this paper, we report (1) the MAE for measuring prediction performance and (2) the realized scheduling cost (as defined in Equation 16) for measuring decision performance. The reason for choosing the MAE is that it allows us to measure errors in stage 1 using the L1-norm, which is thus aligned with stage 2. Note that we measure the *out-of-sample* performance on setting $\mathcal{B}$, that is, how well the approaches generalize to forthcoming order settings. Further, we account for variation in our simulation and thus report results from 10 different runs (i.e., we sample 10 different data sets using the above procedure). We later report the mean and the standard deviation.

5.2 | Baselines

We compare the following approaches for decision making:

- *Naïve machine learning*. Here, we consider off-the-shelf machine learning methods that do not account for distributional shifts. We use two state-of-the-art machine learning methods for comparison: (1) a *regularized linear regression* (i.e., elastic net) as a linear method and (2) a *deep neural network* as a state-of-the-art nonlinear method. Both methods are embedded in our predict-then-optimize framework and thus output scheduling decisions. This allows us to vary the prediction method in stage 1 of our decision problem, while the optimization for job shop scheduling remains identical across all methods. Hence, performance improvements must be attributed to that a method is better in addressing the objective for the predictions in stage 1.

- *Adversarial learning*. This is our proposed approach based on adversarial learning. Here, we use the same predict-then-optimize framework as for the off-the-shelf machine learning baselines. The only difference is that, through the use of adversarial learning, we now account for distributional shifts between order settings and thus directly address the objective of stage 1 in Equation (3). For a fair comparison, we use the *same* model architecture as for the deep neural network in our baselines. This is crucial: It rules out any performance gain due to the larger flexibility of the model. Instead, any performance gain must be solely attributed to the better learning objective.

- *Oracle*. We report an oracle that has access to the ground-truth throughput times without noise. We then use the ground-truth throughput times when solving the optimization problem in Equation (4a)–(4c). Note that the ground-truth throughput times are not available in practice. Instead, the purpose of the oracle is merely to offer a lower bound on the scheduling costs for comparability.

For all methods, we follow common practice (Hastie et al., 2009) and implement a rigorous hyperparameter search via 5-fold cross-validation on the data originating from setting $\mathcal{A}$. Specifically, our hyperparameter search is an exhaustive

TABLE 2 Main results for job shop scheduling.

Approach	Prediction error (MAE)				Scheduling cost			
	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
Linear regression (regularized)	24.3	25.3	28.1	31.5	2989.3	3355.0	3931.0	4453.9
	($\pm$2.0)	($\pm$1.6)	($\pm$1.4)	($\pm$2.1)	($\pm$204.6)	($\pm$250.5)	($\pm$237.8)	($\pm$270.9)
Deep neural network	23.8	24.4	27.2	31.4	2933.1	3209.4	3870.9	4568.4
	($\pm$2.0)	($\pm$1.3)	($\pm$1.5)	($\pm$2.3)	($\pm$216.8)	($\pm$227.9)	($\pm$310.6)	($\pm$293.5)
Adversarial learning (ours)	23.5	23.7	24.0	23.8	2782.2	2802.7	2995.2	2954.2
	($\pm$1.7)	($\pm$1.4)	($\pm$2.1)	($\pm$1.5)	($\pm$182.3)	($\pm$183.9)	($\pm$177.7)	($\pm$176.2)
Oracle (lower bound)	0.0	0.0	0.0	0.0	417.9	503.0	518.4	600.2
	—	—	—	—	($\pm$58.8)	($\pm$77.8)	($\pm$73.9)	($\pm$74.7)

search where all combinations are tested. Details on the hyperparameter tuning procedure can be found in Supporting Information D. For the deep learning neural networks, we follow best practice and use a multilayer feed-forward neural network with rectified linear unit (ReLU) activation and dropout regularization. We use the *same* architecture in our adversarial learning approach. Importantly, we emphasize that our hyperparameter tuning is *fair*. That is, both the deep neural network and our adversarial learning use exactly the *same* tuning grid. Hence, the *same* neural network architecture configurations are tested (i.e., implying that both have the same "budget" for tuning as both have similar runtimes). Hence, since all else is held equal, any improvements must solely be attributed to the fact that one of the two methods has a better objective function in stage 1.

For the optimization, we use the branch-and-cut implementation for mixed-integer problems from the GLPK. All solver parameters are kept at their default values. Note that the optimization is identical for all of the above approaches. Due to computational reasons, we limit the optimization to the 100 production orders with the earliest due date.

5.3 | Results

5.3.1 | Main result with a varying magnitude of the distributional shift

We now report our main results (Table 2). Our results are reported as average values over 10 independent simulation runs ($\pm$ standard deviation).

We make the following observations: (1) Our adversarial learning approach consistently outperforms off-the-shelf machine learning baselines for different magnitudes of the distributional shift in terms of both prediction performance (MAE) and scheduling cost. We also performed a Welch's *t*-test that compares our approach to the better of the two off-the-shelf machine learning baselines. For $\theta = 2, 3, 4$, we find that the improvements in scheduling costs are statistically significant at the 0.1%-significance threshold. (2) The performance gains of our adversarial learning approach become

larger when the magnitude of the distributional shift (θ) is also larger. In other words, the performance of off-the-shelf machine learning quickly deteriorates as the magnitude of the distributional shift is increased, whereas our adversarial learning approach offers substantially more robust performance. For example, for $\theta = 4$, our adversarial learning approach achieves gains in the prediction performance of more than 24% over the baselines, which results in cost savings of more than 33%. Hence, we observe that, across different magnitudes of the distributional shift, lower prediction errors of our adversarial learning lead to better scheduling decisions, ultimately resulting in lower realized scheduling costs. In sum, job shop scheduling using our adversarial learning approach is superior to job shop scheduling using off-the-shelf machine learning by a considerable margin.

5.3.2 | Sensitivity to different capacities and different costs

We now repeat our numerical experiments from above but vary the operational setup (see Figure 8). First, we show the scheduling costs from the above numerical experiment for comparability (left). Second, we use a smaller production line capacity ($K_t = 50$), so that orders have to compete for production lines (center). Third, we use a different cost ratio where we account for the setting where overdue deliveries are more costly than finishing early (right). We thus set cost of tardiness to $c^{\text{tardy}} = 2$ and thus yield a cost ratio $\frac{c^{\text{tardy}}}{c^{\text{early}}} = 2$. All other parameters are identical to the above numerical experiments. The new numerical experiments have only an effect on the scheduling optimization in stage 2 while the predictions from stage 1 are identical to the previous numerical experiments. For that reason, we only report the scheduling costs for varying magnitudes of the distributional shift ($\theta = 1, 2, 3, 4$).

Importantly, the main implications of our main numerical experiment remain unchanged (Figure 8). (1) Our adversarial learning approach outperforms off-the-shelf machine learning baselines. (2) The performance gains from
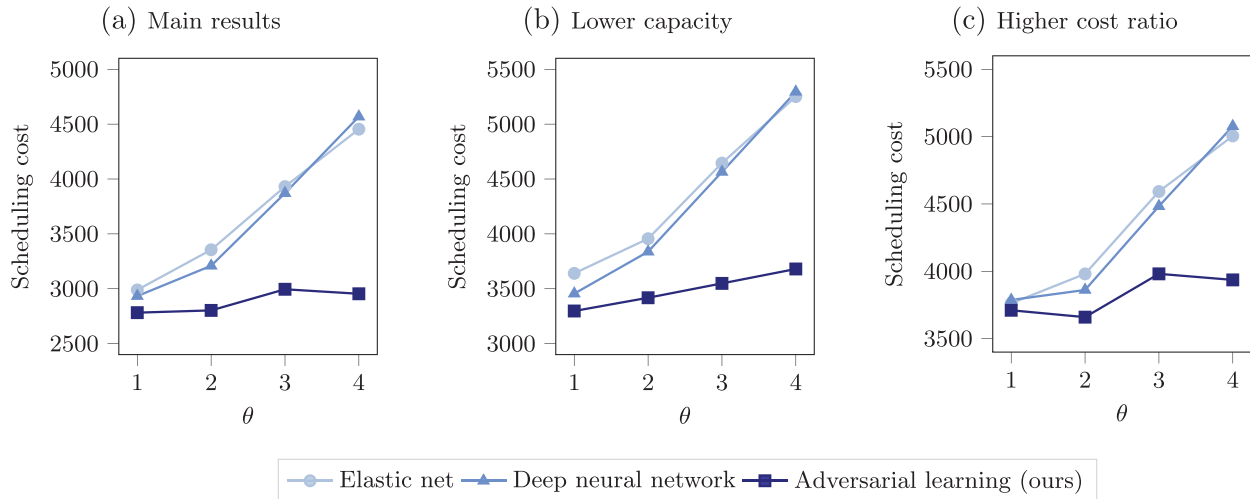
FIGURE 8 Scheduling cost for increasing magnitude of distributional shift across different operational settings. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Results for a different distribution of the error term.

Approach	Prediction error (MAE)				Scheduling cost			
	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
Linear regression (regularized)	25.3	27.6	29.3	33.0	2974.4	3612.1	3983.9	4700.2
	($\pm$1.5)	($\pm$1.6)	($\pm$2.6)	($\pm$2.7)	($\pm$193.1)	($\pm$223.0)	($\pm$338.3)	($\pm$349.2)
Deep neural network	24.4	26.4	28.2	32.2	2891.2	3429.9	3865.4	4637.2
	($\pm$1.5)	($\pm$1.9)	($\pm$2.9)	($\pm$4.0)	($\pm$165.7)	($\pm$326.0)	($\pm$422.1)	($\pm$597.1)
Adversarial learning (ours)	24.2	25.4	25.6	25.8	2741.0	3010.8	3020.3	3030.6
	($\pm$1.3)	($\pm$1.3)	($\pm$1.4)	($\pm$1.2)	($\pm$154.7)	($\pm$274.9)	($\pm$250.7)	($\pm$144.1)
Oracle (lower bound)	0.0	0.0	0.0	0.0	383.4	436.5	476.3	525.7
	—	—	—	—	($\pm$82.5)	($\pm$82.7)	($\pm$85.2)	($\pm$72.2)

our adversarial learning approach increase when the magnitude of the distributional shift is large. We also note that the overall scheduling costs are larger for the numerical experiments with a lower capacity and a higher cost ratio in line with our expectations.

5.3.3 | Sensitivity to error distributions and nonlinearities

We now repeat our main numerical experiment but vary how we generate data in our simulation. In Table 3, we vary the distribution of the error term η in our simulation. We now switch from a Gaussian (in our main numerical experiment) to a uniform distribution, which has shorter tails. The results confirm that our adversarial learning approach outperforms off-the-shelf machine learning baselines. We find this for different magnitudes of the distributional shift in terms of

both prediction performance (MAE) and scheduling costs. In Table 4, we repeat the simulation with a different nonlinearity in the operational data. Here, we replace ϕ with gradient boosting. As above, our adversarial learning approach consistently outperforms the off-the-shelf machine learning baselines for different magnitudes of the distributional shift. We further performed a Welch's t-test that compares the scheduling costs from our approach to the better of the two off-the-shelf machine learning baselines. For $\theta = 2, 3, 4$, the improvements are again statistically significant at common significance thresholds.

Altogether, the results add to the robustness of our adversarial learning approach and demonstrate its operational value for job shop scheduling. Using real-world data from partner company Aker, we find consistent evidence that our adversarial learning approach leads to superior decision making compared to off-the-shelf machine learning (i.e., current industry standard). It can therefore generate substantial cost savings.

TABLE 4 Results for a different nonlinearity in the operational data.

Approach	Prediction error (MAE)				Scheduling cost			
	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
Linear regression (regularized)	23.9	24.6	27.1	30.0	2869.2	3183.2	3726.9	4065.2
	($\pm$1.8)	($\pm$1.3)	($\pm$1.6)	($\pm$1.1)	($\pm$156.5)	($\pm$249.3)	($\pm$234.0)	($\pm$192.5)
Deep neural network	23.6	24.4	27.3	31.2	2823.9	3134.1	3737.7	4360.7
	($\pm$1.9)	($\pm$1.5)	($\pm$2.0)	($\pm$1.7)	($\pm$187.7)	($\pm$243.2)	($\pm$283.0)	($\pm$248.8)
Adversarial learning (ours)	23.4	23.5	24.0	24.2	2697.5	2711.6	2815.0	2835.9
	($\pm$1.7)	($\pm$1.1)	($\pm$1.7)	($\pm$1.5)	($\pm$141.8)	($\pm$156.4)	($\pm$162.4)	($\pm$164.8)
Oracle (lower bound)	0.0	0.0	0.0	0.0	295.4	373.3	362.8	420.8
	—	—	—	—	($\pm$54.2)	($\pm$60.9)	($\pm$58.6)	($\pm$55.8)

6 | ROBUSTNESS CHECKS

6.1 | Machine learning baselines for handling distributional shifts

We consider other baselines from the machine learning literature (Kouw & Loog, 2018; McNamara & Balcan, 2017; Pan & Yang, 2010) that can—in principle—also handle distributional shifts, namely, model retraining and transfer learning with fine-tuning. However, we emphasize that the aforementioned baselines focus on a different setup called supervised domain adaptation (and not unsupervised domain adaptation, as in our decision problem). To this end, baselines from supervised domain adaptation require access to labels from setting $\mathcal{B}$ and are thus only applicable *after* the start of order setting $\mathcal{B}$ and not *before*. This is a crucial difference to our adversarial learning approach, which is designed for operational contexts where labels for the forthcoming order setting $\mathcal{B}$ are absent (i.e., our approach has access to order-specific features $x_i^{\mathcal{B}}$ but *not* to the corresponding labels $y_i^{\mathcal{B}}$). In particular, for job shop scheduling, using model retraining and transfer learning would require access to some of the labels in setting $\mathcal{B}$, which means that one can perform scheduling optimization only *after* production start and hence cannot provide an optimal scheduling sequence for all of the production orders in setting $\mathcal{B}$. Therefore, we assume that the labels of the orders with throughput times that are within the first month of setting $\mathcal{B}$ are known and hence scheduling optimization is done 1 month after the first spools have been produced. The results are shown in Supporting Information F, where we see that these baselines are inferior, despite having access to more information than our approach.

6.2 | Baselines for domain adaptation

As an additional evaluation, we searched the literature for other domain adaptation baselines (for an overview, see Wang & Deng, 2018). Here, another state-of-the-art baseline next to WDGRL is the so-called gradient reversal layer from Ganin et al. (2016). We find that WDGRL is more stable during training compared to a network architecture with a gradient

reversal layer. This is consistent with earlier findings from the machine learning literature (Shen et al., 2018).

6.3 | Robustness in other operational contexts

As an additional robustness check, we consider a different operational context, that is, a different historical manufacturing project at Aker. Thereby, we show that our approach is transferable to other order settings. We again consider two order settings: Johan Sverdrup Living Quarters Rig is used for training in stage 1 of our approach, and Johan Castberg Floating Production Vessel is the order setting for which the job shop scheduling problem is solved. Both order settings do not have chronological overlap. The rest of the experimental setup is identical to Section 5.1. The results yield conclusive findings: Job shop scheduling using an adversarial learning approach is superior over job shop scheduling using off-the-shelf machine learning by a considerable margin (see Supporting Information G).

7 | IMPLICATIONS

7.1 | Methodological implications

To meet order fulfillment targets, manufacturers typically follow a two-staged decision-making process where they first predict the throughput times of production orders and then determine an optimal production schedule. However, predicting throughput times in manufacturing settings with high degrees of product customization is challenging because of distributional shifts between customer orders. Such distributional shifts violate the standard assumption of identically distributed samples in predictive analytics (cf. Hastie et al., 2009), which can harm the prediction performance and thus lead to poor scheduling decisions. To account for distributional shifts, we developed a data-driven approach that combines adversarial learning and job shop scheduling.

In our adversarial learning approach, we make predictions by modeling two adversarial objectives: (1) to predict

throughput times with the best possible prediction performance and (2) to learn a neural network representation that generalizes well across order settings. Specifically, in the latter, we minimize the distance of the neural network representations of the operational data between the historical and the forthcoming order, which reduces prediction errors when applying the model to forthcoming orders with different specifications as the neural network representation is invariant to order settings. As such, the two adversarial objectives force predictions to not be biased toward historical orders but also account for the product specifications of forthcoming orders. This way, we capture distributional shifts and consequently improve decision making in scheduling problems.

While adversarial learning has almost exclusively been applied in computer vision and computational linguistics, this paper analyzes its operational value in an OM problem. As we have shown here, our adversarial learning approach can be effective for manufacturers that produce products with a high degree of customization, and, as such, it overcomes the limitations of existing methods in OM practice. In particular, our approach is different from conventional transfer learning (Kouw & Loog, 2018; McNamara & Balcan, 2017), which requires access to labels of forthcoming orders (i.e., throughput times from the new customer order setting, yet these are only available after completion), whereas our approach circumvents the need for such labels.

7.2 | Managerial implications

As the manufacturing industry is trending toward higher customization (Choi et al., 2021; Feng & Shanthikumar, 2018; Olsen & Tomlin, 2020), decision models that can handle distributional shift gain relevance and importance. For example, Aker stressed the managerial implications of our work: Their business is evolving toward more diverse products, higher volumes, and shorter lead times, which increase the distributional shifts and the relevance of our work for their operational decision making. Hence, managers foresee a larger emphasis on addressing distributional shifts in the future. Yet, issues due to distributional shifts have received limited attention in OM research and practice. Our research contributes insights into how distributional shifts can be detected and provides operations managers with a promising approach to address them. Motivated by our work, we recommend practitioners to more carefully monitor operational data for potential distributional shifts (e.g., via adversarial validation as shown in Section 3.3). For companies, this may serve as an early warning system to identify distributional shifts and inform managers when to take action.

Interestingly, our results suggest that using conventional machine learning—as common in OM practice—has an important limitation. It relies upon the assumption of identically distribution samples and hence cannot account for distributional shifts, which negatively affects prediction performance as well as scheduling costs. As such, OM practitioners must be aware that an off-the-shelf application of popular prediction models, such as deep neural networks, could result in poor decision making. This has direct implications as conventional machine learning models are increasingly used for off-the-shelf predictions in OM practice (Bastani et al., 2022).

Our findings are also relevant beyond manufacturing. Distributional shifts are frequently observed in other areas of management (Simester et al., 2020). In healthcare operations, for example, distributional shifts arise when predicting the mortality risk of rare or new diseases, or when applying machine learning to patient cohorts that are dissimilar from those upon training (Hatt et al., 2022). In marketing, distributional shifts appear when making inferences about customer behavior in emerging segments. Likewise, distributional shifts may also arise for marginalized populations (De-Arteaga et al., 2022). Generally, adversarial learning has the potential to improve managerial decision making in settings subject to extensive heterogeneity.

7.3 | Limitations and opportunities for further research

Our approach relies upon certain technical assumptions, which also hold for most unsupervised domain adaptation algorithms that rely on learning domain-invariant representations (Kouw & Loog, 2018). First, we have assumed a specific form of distributional shift, namely, a covariate shift (Kouw & Loog, 2018), where the distribution of X changes between the two settings $\mathcal{A}$ and $\mathcal{B}$, but the conditional distribution of Y given X remains constant. Covariate shifts imply that the manufacturing processes are comparable across products (i.e., identical specifications lead to the same throughput time regardless of the underlying setting) and are thus common in OM practice. Second, another common assumption in unsupervised domain adaptation is that $P(X)$ has overlapping support between different domains. We have not explicitly made this assumption since the theoretical bounds that motivate our approach do not rely on overlapping support (Ben-David et al., 2007; Redko et al., 2017; Shen et al., 2018). However, many works in unsupervised domain adaptation stress the importance of both assumptions to guarantee successful learning (see Breitholtz et al., 2023; Johansson et al., 2019). Overlapping support should also hold in OM practice as it implies that there is some similarity across orders. Still, decision makers should be careful when using adversarial learning in cases where these assumptions do not hold. We call for further research into these limitations and the development of methods that are robust to them.

7.4 | Concluding remarks

In this paper, we showed that distributional shifts impair decision making in operational settings. As a remedy, we

proposed a data-driven approach combining adversarial learning and job shop scheduling where we address distributional shifts in customized production. Finally, we demonstrated its operational value using a series of numerical experiments based on a real-world job shop production at Aker. An important implication of our work for OM is that both practitioners and researchers need to be aware of potential risks due to distributional shifts in operational settings and, if these occur, must seek effective ways to address them.

DATA AVAILABILITY STATEMENT
Both codes and data are available via a public repository: https://github.com/mkuzma96/CustomProd.

ORCID
Julian Senoner https://orcid.org/0000-0001-6633-5602
Stefan Feuerriegel https://orcid.org/0000-0001-7856-8729

ENDNOTES
[1] Both codes and data are available via a public repository: https://github.com/mkuzma96/CustomProd.

[2] In this paper, we use the mean absolute error (MAE) for measuring prediction performance. This has the benefit that both prediction errors and scheduling costs are measured using the L1-norm; therefore, the costs in stages 1 and 2 are aligned. In principle, other loss functions such as the mean squared error (MSE) could also be used. However, using the MSE is not aligned with the cost penalty from the optimization stage. This is best seen in the following example: A predictive model with a small number of very large errors would have a large MSE but may have a small scheduling cost compared to a predictive with a large number of small errors.

[3] We later also perform a robustness check where we repeat our evaluation with a different ϕ and thus different nonlinearities. Specifically, we use gradient boosting but arrive at a similar conclusion: Our adversarial learning approach remains consistently superior.

REFERENCES
Adams, J., Balas, G., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, *34*(3), 391–401.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial network. In D. Precup, & Y. W. Teh, (Eds.), *Proceedings of the 34th International Conference on Machine Learning (ICML 2017), JMLR.org* (pp. 214–223).

Atan, Z., de Kok, T., Dellaert, N. P., van Boxel, R., & Janssen, F. (2016). Setting planned leadtimes in customer-order-driven assembly systems. *Manufacturing & Service Operations Management*, *18*(1), 122–140.

Baardman, L., Levin, I., Perakis, G., & Singhvi, D. (2018). Leveraging comparables for new product sales forecasting. *Production and Operations Management*, *27*(12), 2340–2343.

Bagchi, U., Julien, F. M., & Magazine, M. J. (1994). Note: Due-date assignment to multi-job customer orders. *Management Science*, *40*(10), 1207–1393.

Ban, G.-Y., Gallien, J., & Mersereau, A. J. (2019). Dynamic procurement of new products with covariate information: The residual tree method. *Manufacturing & Service Operations Management*, *21*(4), 798–815.

Bastani, H. (2021). Predicting with proxies: Transfer learning in high dimension. *Management Science*, *67*(5), 2657–3320.

Bastani, H., Zhang, D., & Zhang, H. (2022). Applied machine learning in operations management. In V. Babich, J. Birge, & G. Hilary (Eds.), *Innovative technology at the interface of finance and operations* (pp. 189–222). Springer.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1-2), 151–175.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In B. Schölkopf, J Platt, & T. Hoffman, (Eds.), *Advances in Neural Information Processing Systems 19, (NIPS 2006). Curran Associates, Inc.* pp. 137–144.

Bernstein, F., Modaresi, S., & Saure, D. (2019). A dynamic clustering approach to data-driven assortment personalization. *Management Science*, *65*(5), 2095–2115.

Bertsimas, D., & Kallus, N. (2019). From predictive to prescriptive analytics. *Management Science*, *66*(3), 1025–1044.

Breitholtz, A., Matsson, A., & Johansson, F. D. (2023). *Unsupervised domain adaptation by learning using privileged information*. https://arxiv.org/pdf/2303.09350

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), 1140–1154.

Chen, X., Owen, Z., Pixton, C., & Simchi-Levi, D. (2022). A statistical learning approach to personalization in revenue management. *Management Science*, *68*(3), 1591–2376.

Choi, T.-M., Kumar, S., Yue, X., & Chan, H.-L. (2021). Disruptive technologies and operations management in the Industry 4.0 era and beyond. *Production and Operations Management*, *31*(1), 9–31.

Choi, T.-M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. *Production and Operations Management*, *27*(10), 1868–1883.

Cohen, M. C. (2018). Big data and service operations. *Production and Operations Management*, *27*(9), 1709–1723.

Cohen, M. A., Ho, T. H., Ren, J., & Terwiesch, C. (2003). Measuring imputed cost in the semiconductor equipment supply chain. *Management Science*, *49*(12), 1653–1670.

Cui, R., Gallino, S., Moreno, A., & Zhang, D. J. (2018). The operational value of social media information. *Production and Operations Management*, *27*(10), 1749–1769.

De-Arteaga, M., Feuerriegel, S., & Saar-Tsechansky, M. (2022). Algorithmic fairness in business analytics: Directions for research and practice. *Production and Operations Management*, *31*(10), 3749–3770.

Federgruen, A., & Mosheiov, G. (1996). Heuristics for multimachine scheduling problems with earliness and tardiness costs. *Management Science*, *42*(11), 1544–1555.

Feldman, J., Zhang, D., Liu, X., & Zhang, N. (2021). Customer choice models versus machine learning: Finding optimal product displays on Alibaba. *Operations Research*, *70*(1), 309–328.

Feng, Q., & Shanthikumar, J. G. (2018). How research in production and operations management may evolve in the era of big data. *Production and Operations Management*, *27*(9), 1670–1684.

Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, *18*(1), 69–88.

Feuerriegel, S., Shrestha, Y. R., von Krogh, G., & Zhang, C. (2022). Bringing artificial intelligence to business management. *Nature Machine Intelligence*, 4, 611–613.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17, 1–35.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553), 452–459.

Glaeser, C. K., Fisher, M., & Su, X. (2019). Optimal retail location: Empirical methodology and application to practice. *Manufacturing & Service Operations Management*, 21(1), 86–102.

Goodfellow, I., Puget-Abadie, J., Mirza, M., Xu, B., Warde-Farlex, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger, (Eds.), *Advances in Neural Information Processing Systems 27 (NIPS 2014). Curran Associates, Inc.* pp. 2672–2680.

Grabenstetter, D. H., & Usher, J. M. (2014). Developing due dates in an engineer-to-order engineering environment. *International Journal of Production Research*, 52(21), 6349–6361.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein GANs. In I. Guyon, U. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett, (Eds.), *Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, Inc.* pp. 5769–5779.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Hatt, T., Tschernutter, D., & Feuerriegel, S. (2022). Generalizing off-policy learning under sample selection bias. In J. Cussens, & K. Zhang (Eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI 2022)* (pp. 769–779). PMLR 180. Association for Uncertainty in Artificial Intelligence.

Huang, T., Bergman, D., & Gopal, R. (2019). Predictive and prescriptive analytics for location selection of add-on retail products. *Production and Operations Management*, 28(7), 1858–1877.

Jakubik, J., & Feuerriegel, S. (2022). Data-driven allocation of development aid toward sustainable development goals: Evidence from HIV/AIDS. *Production and Operations Management*, 31(6), 2739–2756.

Johansson, F. D., Sontag, D., & Ranganath, R. (2019). Support and invertibility in domain-invariant representations. In K. Chaudhuri, & M. Sugiyama (Eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (pp. 527–536) PMLR.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon, U. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran* Associates, Inc., 3149–3157.

Kesavan, S., & Kushwaha, T. (2020). Field experiment on the profit implications of merchants' discretionary power to override data-driven decision-making tools. *Management Science*, 66(11), 5182–5190.

Kouw, W. M., & Loog, M. (2018). *An introduction to domain adaptation and transfer learning*. https://arxiv.org/pdf/1812.11806

Lau, R. Y. K., Zhang, W., & Xu, W. (2018). Parallel aspect-oriented sentiment analysis for sales forecasting with big data. *Production and Operations Management*, 27(10), 1775–1794.

Liu, S., He, L., & Zuo-Jun, M. S. (2020). On-time last mile delivery: Order assignment with travel time predictors. *Management Science*, 67(7), 4095–4119.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2, 56–67.

Mazhar, M. I., Kara, S., & Kaebernick, H. (2007). Remaining life estimation of used components in consumer products: Life cycle data analysis by Weibull and artificial neural networks. *Journal of Operations Management*, 25(6), 1184–1193.

McNamara, D., & Balcan, M.-F. (2017). Risk bounds for transferring representations with and without fine-tuning. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning (ICML 2017), JMLR.org*.

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

Mukherjee, U. K., & Sinha, K. K. (2018). Product recall decisions in medical device supply chains: A big data analytic approach to evaluating judgment bias. *Production and Operations Management*, 27(10), 1816–1833.

Olsen, T. L., & Tomlin, B. (2020). Industry 4.0: Opportunities and challenges for operations management. *Manufacturing & Service Operations Management*, 22(1), 113–122.

Pan, J., Pham, V., Dorairaj, M., Chen, H., & Lee, J.-Y. (2020). *Adversarial validation approach to concept drift problem in user targeting automation systems at Uber by Pan et al. was published at*. In AdKDD 20, which took place on August 23, 2020 San Diego, CA.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.

Redko, I., Habrard, A., & Sebban, M. (2017). Theoretical analysis of domain adaptation with optimal transport. In M. Ceci, J. Hollmén, L. Todorovski, C. Vens, & S. Džeroski (Eds.), Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 737–753), Springer Nature.

Seidmann, A., & Smith, M. L. (1981). Due date assignment for production systems. *Management Science*, 27(5), 493–605.

Senoner, J., Netland, T., & Feuerriegel, S. (2022). Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing. *Management Science*, 68(8), 5557–6354.

Shalit, U., Johansson, F. D., & Sontag, D. (2017). Estimating individual treatment effect: Generalization bounds and algorithms. In D. Precup, & Y. W. Teh (Eds.) *Proceedings of the 34th International Conference on Machine Learning* (pp. 3076–3085).

Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. In S. A. McIlraith, & K. Q. Weinberger, (Eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)* AAAI Press, Palo Alto, CA, pp. 4058–4065.

Simester, D., Timoshenko, A., & Zoumpoulis, S. I. (2020). Targeting prospective customers: Robustness of machine-learning methods to typical data challenges. *Management Science*, 66(6), 2495–2522.

Song, J.-S., Xu, S. H., & Liu, B. (1999). Order-fulfillment performance measures in an assemble-to-order system with stochastic leadtimes. *Operations Research*, 47(1), 131–149.

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In R. Chellappa, A. Hoogs, & Z. Zhang, (Eds.), *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), IEEE Computer Society* Los Alamitos, CA, pp. 2962–2971.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using *t*-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.

Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312, 135–153.

Wang, Q., Michau, G., & Fink, O. (2019). Domain adaptive transfer learning for fault diagnosis, 2019. In *Prognostics and System Health Management Conference* pp. 279–285. Publisher: IEEE. No editors available. Event: Prognostics and System Health Management Conference (PHM-Paris 2019), Paris, France, May 2–5, 2019.

Wein, L. M., & Chevalier, P. B. (1992). A broader view of the job-shop scheduling problem. *Management Science*, 38(7), 1018–1033.

Yoon, J., Jordon, J., & van der Schaar, M. (2018). GANITE: Estimation of individualized treatment effects using generative adversarial nets. In I.

Murray, M. Ranzato, O. Vinyals, T. Sainath, Y. Bengio, & Y. LeCun, (Eds.), *Proceedings of the International Conference on Learning Representations. Could not find page range. Conference was held in Vancouver, Canada, Mon Apr 30th through May 3rd, 2018.*

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.