A META ANALYSIS OF DATA-DRIVEN NEWSVENDOR APPROACHES

Simone Buttler*, **Andreas Philippi***, **Nikolai Stein**, **Richard Pibernik**
Chair of Logistics and Quantitative Methods
Julius Maximilian University of Würzburg
Würzburg, Germany
simone.buttler@uni-wuerzburg.de

ABSTRACT

Recently, a number of publications in leading operations management and operations research journals proposed new models that combine machine learning and mathematical optimization techniques to predict inventory decisions directly from historical demand and additional feature information. This paper presents the results of a meta analysis of recent machine learning based approaches for solving the most prominent problem in operations management, the newsvendor problem. We find that the reproducibility of existing studies is typically low because authors evaluate their new approaches based on small and proprietary datasets, do not share data and code, and use different benchmarks. We develop a reproducible, unified evaluation procedure and apply various models to a large and heterogeneous dataset. Our results do not support the findings and claims of most of the recent papers, and, in several cases, we even obtain contradicting results. In general, the robustness of the newly proposed models appears to be low. To support both researchers and practitioners in the development and evaluation of new models, we provide extensive benchmark data and a Python library that contains open source implementations of most existing models.

1 INTRODUCTION

In today's fast-paced world, companies face considerable uncertainty when making important decisions in operations management, for example, when deciding on capacity, inventory levels, transportation, and production schedules. However, with the rise of digitization, companies have gained unprecedented access to data related to their particular decision problem, offering the opportunity to reduce the degree of uncertainty. For example, in inventory management, the decision maker may have access to historical demand data, as well as additional side information that may be predictive of uncertain demand, such as weather data, calendar information, and data extracted from social media. Driven by the availability of such rich data sources, a stream of literature in operations management research called data-driven operations management (DDOM) has recently emerged. Newly proposed DDOM approaches depart from the traditional predict-then-optimize (PO) paradigm that has been standard in operations management: Instead of first predicting the uncertain variable of interest and then solving a stochastic optimization problem, they integrate machine learning and optimization techniques to *directly* predict a cost-optimal decision from historical data.

As the classical single period inventory management setting, the newsvendor problem naturally became a starting point for developing DDOM approaches. In the newsvendor problem, a decision maker has to determine the optimal inventory quantity for a single period and incurs costs when demand is higher or lower than the inventory. The problem is different from "standard" regression problems studied in machine learning research in two ways: First, the loss function is non-symmetrical as over-predicting (i.e., having too much stock at hand) may incur different costs than under-predicting (i.e., not being able to fulfill all customers' demands). Second, identical absolute errors, in terms of units ordered, may not be of the same importance for all instances, because different products make different contributions to a company's bottom line. Therefore, models must be

[*] Authors contributed equally

evaluated on the basis of the bottom-line impact and not on their predictiveness. Recently, a number of new DDOM approaches have been published to solve the news vendor problem in leading operations management and operations research journals. The common denominator of these papers is that they propose and analyze new DDOM approaches and demonstrate that they outperform some benchmark model(s). However, the reproducibility of these studies is typically low, because most authors only evaluate their approaches based on small and proprietary datasets, and do not share data and code. In addition, they use a variety of different benchmarks. This renders a comprehensive comparison of the different DDOM models impossible and creates problems for researchers and practitioners: Researchers have no objective means to benchmark against state of the art (SOTA) approaches, and practitioners can hardly identify the best approach for solving their real-life problems in a robust fashion.

In this paper, we perform a meta analysis of existing DDOM models for solving the newsvendor problem on large and very heterogeneous data using a standardized and reproducible procedure. In our study, we cannot reproduce the results of most of the previous papers and, in a number of cases, we even obtain contradictory results. We find that model robustness is low and that model performance depends on the specific product for which an inventory decision has to be taken, the parameters of the loss function, and the available feature information. More specifically, we observe that any one of the models under consideration can, under certain conditions, be optimal for an individual product—there is no dominating approach that can be established as SOTA. Next to these important insights, we make several other contributions that can enhance future research on DDOM models: (1) We are the first in the field of DDOM to provide extensive benchmark data for the newsvendor problem. (2) We developed an open source python package providing access to the most relevant DDOM models to allow easy and efficient performance comparison and benchmarking. (3) We provide guidance on how to carry out an objective, structured, and reproducible evaluation of DDOM approaches that can also be applied to problems other than the newsvendor problem.

2 The newsvendor and data-driven solution approaches

Among the many inventory control problems that have been addressed in the OM literature, the newsvendor problem is the most basic single period inventory problem under demand uncertainty. As such, it became the natural starting point for developing DDOM approaches (Qi et al., 2020). In a newsvendor setting, the decision maker decides on the inventory of a single product for a single selling season. Any leftover demand at the end of the season leads to overage costs of c_o per unit. The decision maker incurs underage costs c_u for each unit of demand that cannot be satisfied. Consequently, the decision maker seeks to determine the order quantity q that minimizes the total expected costs. For a single product, the problem can be stated as follows:

$$\min_{q \geq 0} = \mathbb{E}_D[c_u(D-q)^+ + c_o(q-D)^+], \tag{1}$$

where D is the random demand, and $(\cdot)^+ := \max\{0, \cdot\}$. If the demand distribution F is known, the optimal solution to (1), denoted by q^*, is given by the $c_u/(c_u + c_o)$ quantile:

$$q^* = F^{-1}\left(\frac{c_u}{c_u + c_o}\right), \tag{2}$$

where F^{-1} is the inverse cumulative density function (cdf) of D. In practice, the decision maker cannot directly solve Equation 2, because he does not know the true distribution of D. However, historical demand data $d_1, ..., d_n$ are often available that can be used to solve the empirical counterpart of Equation 1:

$$q^* = \min_{q \geq 0} \frac{1}{n} \sum_{i=1}^{n} \left[c_u \left(d_i - q\right)^+ + c_o \left(q - d_i\right)^+ \right] \tag{3}$$

The literature refers to this approach as sample average approximation (SAA) (Levi et al., 2015). In today's data-rich environments, companies have access not only to historical demand observations, but also to potentially large datasets $S_n = \{(d_1, \boldsymbol{x}_1), \ldots, (d_n, \boldsymbol{x}_n)\}$ that contain historical demand observations d_t and corresponding feature vectors $\boldsymbol{x}_t (t = 1, ..., n)$. The elements of the feature vectors can be any type of information that may be predictive of the uncertain demand. The new DDOM approaches addressed in this paper learn a decision function from S_n that predicts an inventory decision $q(\boldsymbol{x})$ for each new observation $\boldsymbol{x}$. The existing approaches can be classified into function approximation approaches that are based on the principle of empirical risk minimization (ERM), and approaches that integrate empirical conditional density estimation and optimization.

2.1 Empirical Risk Minimization-based Approaches

The approaches contained in this first class seek to learn a function $q(\cdot) : \mathcal{X} \to \mathcal{Q}$ that maps directly from the feature space $\mathcal{X}$ to a decision space $\mathcal{Q}$ by minimizing the empirical risk, which is defined as the average cost over the training data S_n. More formally, the problem to be solved is given by:

$$\min_{q(\cdot)\in\mathcal{F}} R_N(q(\cdot); S_n) := \frac{1}{n}\sum_{i=1}^{n}\big[c_u(d_i - q(\boldsymbol{x}_i))^+ + c_o(q(\boldsymbol{x}_i) - d_i)^+\big], \tag{4}$$

where R_N is the empirical risk of $q(\cdot)$ and $\mathcal{F}$ is a function space. Given the function $q(\cdot)$, one can directly determine a decision $q(\boldsymbol{x})$ for each new observation $\boldsymbol{x}$. The solution to Equation 4 is equivalent to the solution of a high-dimensional quantile regression. To learn $q(\cdot) : \mathcal{X} \to \mathcal{Q}$, several different machine learning methods have been used in the literature. Beutel & Minner (2012) and Ban & Rudin (2019) restrict $\mathcal{F}$ to the space of linear functions. Oroojlooyjadid et al. (2020) and Huber et al. (2019) allow for a non-linear function space and determine $q(\cdot)$ by training deep neural networks that minimize the empirical risk in Equation 4. In the remainder of the paper, we use the acronyms LR to refer to the linear models proposed by Beutel & Minner (2012) and Ban & Rudin (2019), and DL to refer to the models proposed by Oroojlooyjadid et al. (2020) and Huber et al. (2019).

2.2 Conditional Density Estimation and Optimization

The approaches contained in this second class are based on deriving some data-driven sample weights from features and optimizing SAA against a re-weighting of the data, as expressed in Equation 5:

$$q^*(\boldsymbol{x}) = \arg\min_{q\in\mathcal{Q}} \sum_{i=1}^{n} w_i(\boldsymbol{x})\big[c_u(d_i - q)^+ + c_o(q - d_i)^+\big], \tag{5}$$

where $\boldsymbol{x}$ is the feature vector of a new instance and $w_i(\cdot)$ is a function that assigns a weight $w_i \in [0, 1]$ to each sample $(d_i, \boldsymbol{x}_i)$ based on the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}$. We refer to this approach as conditional density estimation and optimization (CDEO). The optimal solution to (5) is

$$q^*(\boldsymbol{x}) = \inf\left\{ q : \frac{\sum_{i=1}^{n} w_i(\boldsymbol{x})\mathbb{1}(d_i \le q)}{\sum_{i=1}^{n} w_i(\boldsymbol{x})} \ge \frac{c_u}{c_u + c_o} \right\}. \tag{6}$$

In contrast to the ERM-based approaches, the CDEO approaches define $q(\boldsymbol{x})$ point-wise. Thus, one has to first determine the sample weights w_i for each new instance $\boldsymbol{x}$, before solving the optimization problem in (5). Obviously, the performance of CDEO is driven by the way the sample weights are calculated. Multiple weight functions have been proposed in the literature, in particular by Bertsimas & Kallus (2020), who construct a number of different weight functions based on a variety of predictive machine learning methods, including k-nearest-neighbors, decision tree, and random forest regression. We refer to these approaches as KNNW, DTW, and RFW. Moreover, both Bertsimas & Kallus (2020) and Ban & Rudin (2019) propose to use kernel weight functions (KW).

2.3 Review of competing data-driven models

In Table 1 we give an overview of the relevant papers that proposed DDOM approaches applicable to the newsvendor problem, including the datasets and benchmarks that were used for evaluation purposes. While all models have shown superior performance compared to some benchmarks, under certain conditions, it still remains unclear how to compare their performances. The reason for this is twofold: On the one hand, most researchers use their own proprietary dataset for evaluation. The datasets vary in terms of the domain from which the data are drawn, the number and type of features, and the length of the corresponding time series. On the other hand, researchers use different models for benchmarking, making a direct comparison of results impossible. For example,

Table 1: **Overview of relevant DDOM papers** (* denotes the models included in our evaluation)

Paper	Model	Benchmark	Data
Beutel & Minner	LR	OLS, MM	Proprietary retail chain data of 64 stores for 270 days; feature information includes price, weather data, and weekdays.
Ban & Rudin	LR*, KW*	PO, SAA	Proprietary data for a nurse staffing problem including demand for nurses in a hospital for 2644 time periods; feature information includes calendar data as well as lags.
Bertsimas & Kallus	KW, KNNW*, DTW*, RFW*	SAA, PP	Proprietary media vendor data including demand for various items and locations for 150 weeks; feature information includes 91 features with information on items, locations, dates, lags, as well as social media data.
Oroojlooyjadid et al.	DL*	KNNW, KW, RFW, LR, PO	Extract of Pentaho MySQL Foodmart Database including retail data from 24 different departments; feature information includes calendar data.
Huber et al.	DL	PO, LR	Proprietary bakery chain data of eleven products for five stores for 528 days; feature information includes calendar data, weather data, and locations of the stores.

Beutel & Minner (2012) use retail chain data and benchmark their LR model against ordinary least squares (OLS) regression and the method of moments (MM). Ban & Rudin (2019) use empirical data from a nurse staffing problem to evaluate their models and show that both LR and KW outperform traditional PO approaches as well as SAA. Bertsimas & Kallus (2020) use data provided by a big international media vendor for testing. As benchmarks, they use random forest point-predictions and SAA, and show that their RFW model performs best. Building upon the work of Bertsimas & Kallus (2020) and Ban & Rudin (2019), Oroojlooyjadid et al. (2020) show that their DL model not only outperforms traditional PO approaches, but also existing data-driven models, including LR, KW, KNNW, and RFW when applied to their extract of the Pentaho MySQL Foodmart Database. Huber et al. (2019) use proprietary data of a bakery chain to compare, among others, an LR and a DL approach to their PO counterparts. Their results suggest that for limited data availability the traditional PO approaches outperform their DDOM counterparts. The performance of DDOM models increases with data availability in terms of features and historical observations. They do not find any DDOM model that consistently outperforms conventional PO approaches.

In our study we include the LR and the KW model of Ban & Rudin (2019), because their paper is published in the highest ranking journal of the discipline and has been the first and most extensive paper on the "Big Data Newsvendor". We also include the KNNW, DTW, and RFW models of Bertsimas & Kallus (2020), as they provide the most extensive analysis and evaluation of CDEO methods. From the two similar papers proposing DL approaches, we select Oroojlooyjadid et al. (2020), because their implementation is similar to that of Huber et al. (2019), but in their evaluation, their model outperformed all other DDOM models, including those of Bertsimas & Kallus (2020); Ban & Rudin (2019).

3 Experimental Setup

In the previous section, we briefly outlined that existing papers claim that their models outperform one or more benchmark approaches, but that these claims can hardly be verified beyond individual datasets and particular benchmarks. The main goal of our study is an objective and fair performance comparison of the different models—we want to evaluate their robustness, validate the claims made in previous papers, and ascertain whether there is a model that can be recommended as SOTA. To compare the models, we propose a reproducible, unified procedure that is based on standards established in the machine learning community. We share code and data via our GitHub repository [1] to

[1] https://github.com/opimwue/A-structured-evaluation-of-data-driven-newsvendor-approaches

make our experiments transparent and reproducible. We developed the open-source Python package *ddop* (Philippi et al., 2021), to provide easy and efficient access to the data-driven newsvendor models discussed in Section 2.

3.1 Data and Experiments

In our experiments, we use four heterogeneous datasets (Bakery, Restaurant, subset of M5, Store Item Demand (SID)). An overview of the datasets is provided in Table 2. More detailed information is provided in Appendix A. As an important step towards reproducibility, we make all of the datasets available [2]. The datasets cover various domains, are of different sizes in terms of the number of products and the length of the time series, and include different features.

Table 2: **Overview of dataset**

Dataset	Domain	Products × Shops	N	Features
Bakery	Bakery	3 × 5	1215	calendric, lag, weather, holidays
Restaurant	Restaurant	7 × 1	765	calendric, lag, weather, holidays, promotions
M5 (subset)	Retail chain	10 × 10	1942	calendric, lag, events
SID	Retail chain	50 × 10	1826	calendric, lag

To evaluate the robustness of the models in our numerical experiments, we follow a fractional factorial design and vary two specific dimensions: loss function and features.

Loss function: In most machine learning problems, a static loss function is used to evaluate the performance of different predictive models, e.g., the MSE or RMSE for regression problems. This is different in an operations management context. The loss function measures real-world costs and, therefore, depends on the parameters that influence these costs. In our newsvendor setting, the loss function is determined by c_u and c_o as defined in Equation 1 and may be asymmetric, depending on the particular values of these parameters. The relationship between c_u and c_o is captured by the so called service level (sl), defined as $c_u/(c_u + c_o)$ (see Equation 2). Naturally, a good model should be robust across different service levels so that it can be used in various contexts—even within one company the relationship between c_u and c_o can differ depending on the product or the selling season.

Features: Feature availability may vary across practical problems. Therefore, it is important to evaluate the robustness of different models with respect to the feature information. In our evaluation, we define three different feature categories: calendric, lag, and special features. Both calendric and lag features are used to capture the characteristics of the demand time series. Calendric features contain only the information that can be extracted directly from the date of the time series (e.g., weekdays, month, year). Lag features capture information from previous time periods, such as past demand observations, allowing the models to learn properties of the time series, such as trend and seasonality[3]. In contrast to the first two feature categories, the special features depend on the respective dataset and include domain knowledge, such as information about promotions, special events, or weather conditions.

We define three different experiments (see Table 3). We begin with a base case scenario that resembles a typical industry setting—90% service level (sl) with only calendric and lag features. Then we vary sl reflecting the ratio of the parameters c_u and c_o of the loss function. Finally, we vary the feature information to assess the models' robustness toward different levels of feature availability.

[2] While M5 and SID were published on Kaggle as part of a competition, the other two datasets were provided by our industry partners and are accessible via our GitHub repository

[3] To generate lag features, we use the Python library tsfresh (https://tsfresh.readthedocs.io/en/latest/index.html). More specifically, we compute basic descriptive statistics (e.g., minimum, maximum, mean, variance) for three different rolling windows of length 7, 14, and 28 days.

Table 3: **Overview of experiments**

	Loss Function	Features	Results
Base scenario	sl = 0.9	X = [calendar, lag]	Section 4.1
Loss function variation	sl $\in \{0.1, 0.25, 0.5, 0.75, 0.9\}$	X = [calendar, lag]	Section 4.2
Feature variation	sl = 0.9	X $\in$ {[calendar], [calendar, lag], [calendar, lag, special]}	Section 4.2

3.2 Evaluation process

For each experiment, we first group each dataset by product and store—we call a single product-store combination an instance. Subsequently, we apply each of the following steps for each instance, service level, and feature category.

Transformation: We apply one-hot-encoding to transform all categorical features into their binary representation.

Train-test split: We split the data into a train set containing 75% of the data and a test set containing the remaining 25%. We do not apply shuffling to preserve the structure of the time series.

Scaling: We apply standardization (removing the mean and scaling to the unit variance) to all continuous features using the scikit-learn standard scaler (Pedregosa et al., 2011). More specifically, we fit the scaler to the training data and then transform both the train and the test set.

Model training: For each model that provides hyper-parameters, we apply a grid search on the train set with 10-fold cross validation to find the best parameters leading to the lowest average cross-validation cost. Subsequently, we fit the model to the entire train set. We provide the hyperparameter grids specific to each model in Appendix C.

Model evaluation: To enable an objective comparison across datasets, we require the metrics used to be relative measures that operate on a universal scale. Therefore, we cannot use the empirical costs on the test set directly, but normalize them by the SAA costs. SAA is a natural baseline as it does not include feature information. Given a model k we compute the cost delta to SAA ΔC_k as follows:

$$\Delta C_k = 1 - \frac{R_{N_{test}}(q_k, S_{test})}{R_{N_{test}}(q_{SAA}, S_{test})} \tag{7}$$

where R_N is the empirical risk as defined in Equation 4.

Statistical significance test: In the course of our numerical evaluation, we also want to test our findings for statistical significance. In particular, differences in results obtained from model recommendation versus model selection. We use the one-sided Wilcoxon's signed-rank test (Wilcoxon, 1945) that computes the paired differences between samples $d_i = a_i - b_i$ and tests whether $median(d) > 0$. In our case, whether the difference in relative performance improvement over SAA between model selection and model recommendation is greater than zero. It does not assume data to be normally distributed. We test for different levels of significance.

4 Results

This section presents the results of our numerical analysis, as described in Table 3. We first evaluate and discuss the models' performance and robustness across the different datasets in the base case. Then, in Section 4.2, we evaluate how robust the models are towards variations of the parameters of the loss function, and feature availability. Finally, in Section 4.3, we evaluate how robust the models are relative to a model selection approach—that is, to an approach that selects, a priori, the best model for each instance, based on cross-validation.

4.1 Robustness analysis across datasets

We first explore the performance and robustness of the different models in the base case setting. Figure 1 presents the relative performance improvement over SAA for all models by dataset. The performances of the different models and their rank order, based on the mean performance, vary across datasets. RFW and LR lead to highest mean performance improvements in three out of the four datasets. In the Restaurant dataset, RFW leads to the highest mean performance, followed by DL. It is surprising that, in most cases, the relatively simple LR model leads to a better performance than models that can account for non-linear relationships between features and the decision variable. The results are not in line with those of Ban & Rudin (2019), where LR was inferior compared to KW, and they do not support the findings in Oroojlooyjadid et al. (2020) where DL consistently outperformed all other models. Interestingly, in their study LR was the worst performing model, with average cost increases of 53% compared to DL. Our results are clearly contradicting these findings. Based only on the rank order per dataset, either RFW or LR should be the models of choice. However, the overlapping error bars in Figure 1 suggest that this conjecture may not hold for individual instances. We carried out additional analysis on an instance-level and found that in none of the datasets there is a single model that dominates all others (see the base case results in Figure 2); more importantly, we also observe that there is no dominated model—that is, each of the models considered in our study leads to the best performance for at least one instance. In Section 4.3 we address the performance impact of choosing the "optimal" model for each instance.

The discussion of the results of our base case analysis reveals two interesting insights: We are not able to reproduce the results obtained in previous studies, and there is no model with a robust performance across all datasets and instances.

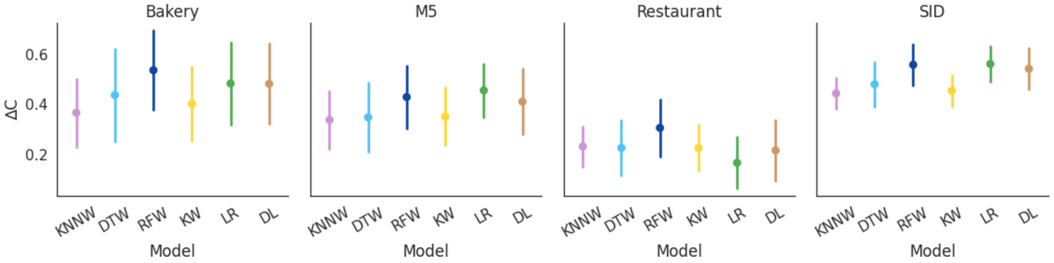

Figure 1: **Cost delta to SAA for base case.** Presenting the results for each model by dataset. Averages across instances along with their standard deviation are presented.

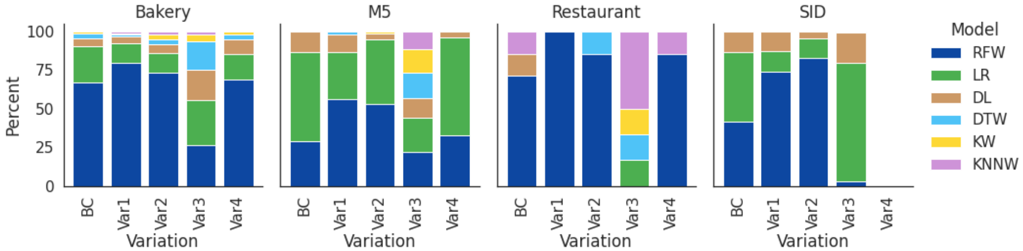

Figure 2: **Breakdown of optimal models.** Presenting the share of instances for which each model type is optimal by dataset and across different experiment parameter variations. BC (base case): sl = 0.9, X = [calendar, lag]; Var1: sl = 0.75, X = [calendar, lag]; Var2: sl = 0.5, X = [calendar, lag]; Var3: sl = 0.9, X = [calendar]; Var4: sl = 0.9, X = [calendar, lag, special]

4.2 Robustness analysis depending on the loss function and feature availability

We now assess the models' robustness towards variations of the parameters of the loss function (reflected by the service level (sl)) and the availability of features X. To this end, we compare the breakdown of the optimal models in the base case to the breakdown under different service levels and feature categories. Selected results are displayed in Figure 2 and the detailed results are provided in Appendix B. The results in Figure 2 suggest that the choice of the optimal model strongly depends on both the parameters of the loss function and the features considered. In the restaurant dataset, for example, we see that the share of the optimal models depends strongly on the feature availability (compare Figure 2 BC, Var3, Var4). From the results in the SID dataset, we observe that the share of optimal models strongly depends on the parameters of the loss function (compare Figure 2 BC, Var1, Var2). These results support our initial conjecture in Section 4.1 that there is no single model that is robust across all datasets and instances under varying parameters of the loss function and feature availability. Of course, in this section we only consider the optimality of different models. In the next section, we address the robustness of the different models in terms of their cost performance.

4.3 Robustness analysis of models' cost performance

To evaluate the robustness in terms of cost performance, we benchmark the individual models against a model selection approach. In the model selection approach, we identify the best model for each instance based on cross-validation. The difference between the performance of the model selection approach and the performance of an individual model serves as a measure of the model's performance robustness. In Figure 3, we report the pairwise cost difference between the model selection approach and the individual models for each instance. First of all, we see that in many cases the model selection approach leads to a statistically significant and substantial performance improvement compared to fixing one model a priori. However, the results are strongly dependent on the individual datasets. Although the results displayed in Figure 2 suggest a high value of model selection for M5 and SID, we see that we do not experience a large performance impact from model selection. Choosing the LR approach a priori only induces small performance losses that are not statistically significant in the SID dataset. On the contrary, in the other two datasets, fixing LR a priori has a very strong and significant detrimental impact on performance. Similarly, the RFW model performs well in the Bakery and Restaurant dataset but significantly worse than model selection in M5 and SID. The results of this analysis displayed in Figure 3 support our initial hypothesis that there is no model that can be recommended as SOTA and that the robustness of the individual models is low.

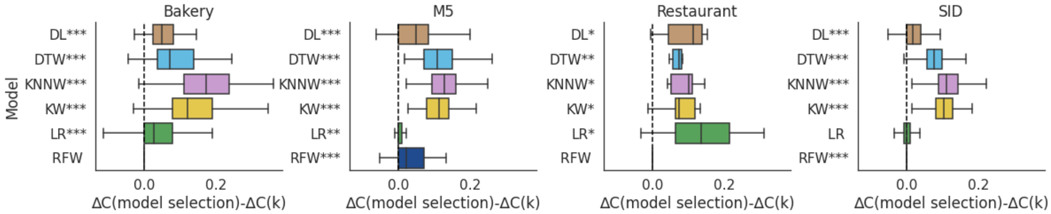

Figure 3: **Performance comparison of model selection versus fixing the model type a priori.** Presenting the difference between the cost delta to SAA of model selection and using a selected model k. Boxplots across instances are shown for the base case scenario. Wilcoxon's signed-rank test results: *: p<0.1, **: p<0.01, ***: p<0.001

5 Summary and Discussion

Recently, a number of papers have been published in leading operations management and operations research journals proposing new machine learning based approaches for solving the newsvendor problem, the most prominent optimization problem in operations management. A key motivation of our study is that the reproducibility of these studies is typically low: They use proprietary data, different benchmarks, and do not share data and code. We are the first to conduct an extensive meta-analysis of these approaches. Our evaluation is based on large and heterogeneous data and a unified evaluation procedure. Our findings do not support a number of claims made in previous papers with respect to the performance of the newly proposed methods. In some cases, we even obtain results that strongly contradict the conjectures made by the authors. Our results suggest that the performance robustness of the new methods is low. There is neither a dominant model nor a model that is always dominated. This indicates that an evaluation of established and new methods requires a standardized evaluation procedure, a large set of relevant benchmark data, and standards for sharing model implementations that enable reproducible comparisons. We take a first step in this direction by: (1) providing extensive benchmark data for the research community, (2) outlining an evaluation procedure that allows for a fair and objective comparison of alternative approaches, and (3) providing the open source Python library *ddop* (`https://github.com/opimwue/ddop`) that enables an easy and efficient comparison of different approaches.

References

Gah-Yi Ban and Cynthia Rudin. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019.

Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.

Anna-Lena Beutel and Stefan Minner. Safety stock planning under causal demand forecasting. *International Journal of Production Economics*, 140(2):637–645, 2012.

Jakob Huber, Sebastian Müller, Moritz Fleischmann, and Heiner Stuckenschmidt. A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278(3): 904–915, 2019.

Retsef Levi, Georgia Perakis, and Joline Uichanco. The data-driven newsvendor problem: new bounds and insights. *Operations Research*, 63(6):1294–1306, 2015.

Afshin Oroojlooyjadid, Lawrence V Snyder, and Martin Takáč. Applying deep learning to the newsvendor problem. *IISE Transactions*, 52(4):444–463, 2020.

Fabian Pedregosa, Gaë Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Andreas Philippi, Simone Buttler, and Nikolai Stein. ddop: A python package for data-driven operations management. *Journal of Open Source Software*, 6(66):3429, 2021.

Meng Qi, Ho-Yin Mak, and Zuo-Jun Max Shen. Data-driven research in supply chain operations-a review. *Available at SSRN*, 2020.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80, 1945.

A Datasets

Bakery: This dataset is provided by a local bakery chain. The dataset contains sales data for three different products at 35 different stores over a period of 1215 days. Every evening, each store must order the products from a central factory that are delivered the next morning. Reordering during the day is not possible. Unsold goods have to be disposed of at the end of the day. Thus, the problem at hand can be considered a newsvendor problem. All products are everyday items with typically high stock levels making censored demand unlikely. We exclude 10 unique product-store combinations due to demand intermittency. Next to calendric and lag features, the dataset contains information on weather, promotions, and holidays.

Restaurant: This dataset contains daily sales data from a restaurant for 7 different main ingredients on 765 days. To prepare the meals, the restaurant must decide how much of the ingredients to defrost overnight. The defrosted ingredients must then be sold within the next day. Leftovers are disposed of. Thus, the problem at hand can be considered a newsvendor problem. During data recording, the store manager's strategy was to maintain a service level of almost 100%, which is why we consider censored demand not to be an issue. Next to calendric and lag features, this dataset contains weather as well as special features.

M5: The M5 dataset[4] contains daily sales from Walmart stores and was made available as part of the well known forecasting competition M5. The original dataset contains sales records for 3,049 products across 3 product categories, 7 departments, and 10 stores on 1942 days. For our analysis, we select only data that belongs to the product category ”Foods” as this is most relevant in a newsvendor setting. To avoid intermittent demand, we select only the top 10 products which exhibit the least intermittency. This leads us to 100 unique product store combinations selected for the numerical evaluation. Next to calendric and lag features, this dataset contains features that indicate special events such as sporting events or payout days.

SID: The store item demand dataset was made available as part of the Kaggle Store Item Demand Forecasting Challenge[5]. It contains sales data from 50 different products in 10 different stores for 1826 days. It is our largest dataset in terms of unique product store combinations but contains only calendric and lag features derived from the time series.

B Relative performance improvement over SAA under variation of service levels and features

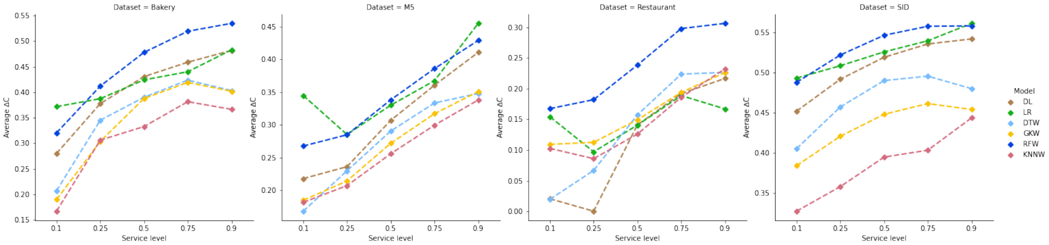

Figure 4: Average cost delta to SAA across service levels and datasets

[4] https://www.kaggle.com/c/m5-forecasting-accuracy

[5] https://www.kaggle.com/c/demand-forecasting-kernels-only

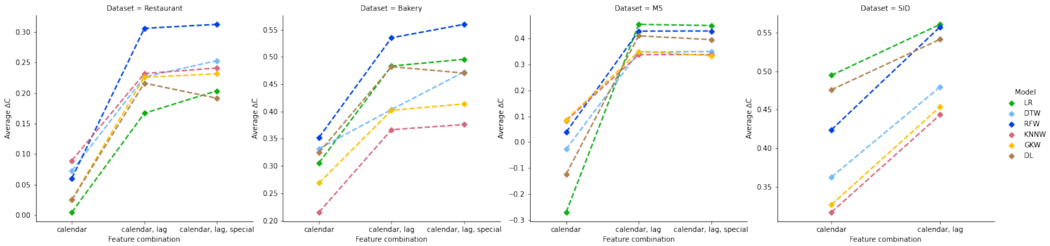

Figure 5: Average cost delta to SAA dependent on feature availability

C Hyperparameter grids

The section states the list of hyperparameters tuned including their search spaces. If a model is not listed here, it has no hyperparameters to tune.

- DTW:
  - Maximum depth of the tree: [None, 2, 4, 6, 8, 10]
  - Minimum number of samples required to split an internal node: [2, 4, 6, 8, 16, 32, 64]
- RFW:
  - Maximum depth of a tree: [None, 2, 4, 6, 8, 10]
  - Minimum number of samples required to split an internal node: [2, 4, 6, 8, 16, 32, 64]
  - Number of trees in the forest: [10, 20, 50, 100]
- KNNW:
  - Number of neighbors to use: [1, 2, 4, 8, 16, 32, 64, 128]
- KW:
  - Kernel bandwidth: [0.5, 0.75, 1, $\ldots$, $\lceil\sqrt{n_{features}/2}\rceil + 0.25$]
- DL:
  - Optimizer: [”adam”]
  - Network architecture (number of neurons in each layer):
    - ◇ $(\lceil 0.5 \cdot n_{features} \rceil,\ \lceil 0.5 \cdot 0.5 \cdot n_{features} \rceil)$
    - ◇ $(\lceil 0.5 \cdot n_{features} \rceil,\ \lceil 0.5 \cdot 1 \cdot n_{features} \rceil)$
    - ◇ $(1 \cdot n_{features},\ \lceil 0.5 \cdot 1 \cdot n_{features} \rceil)$
    - ◇ $(1 \cdot n_{features},\ 1 \cdot 1 \cdot n_{features})$
    - ◇ $(2 \cdot n_{features},\ \lceil 2 \cdot 0.5 \cdot n_{features} \rceil)$
    - ◇ $(2 \cdot n_{features},\ 2 \cdot 1 \cdot n_{features})$
    - ◇ $(3 \cdot n_{features},\ \lceil 3 \cdot 0.5 \cdot n_{features} \rceil)$
    - ◇ $(3 \cdot n_{features},\ 3 \cdot 1 \cdot n_{features})$
  - Epochs: [10, 100, 200]