ARTICLE IN PRESS

International Journal of Forecasting xxx (xxxx) xxx

Contents lists available at ScienceDirect

International Journal of Forecasting

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

Machine learning for satisficing operational decision making: A case study in blood supply chain

Mahdi Abolghasemi [a,*], Babak Abbasi [b], Zahra HosseiniFard [c]

[a] School of Mathematics and Physics, The University of Queensland, St Lucia, QLD 4067, Australia
[b] Department of Information Systems and Business Analytics, RMIT University, Melbourne, VIC 3000, Australia
[c] Faculty of Business and Economics, The University of Melbourne, Parkville, VIC 3010, Australia

ARTICLE INFO

Keywords:
Forecasting
Constraint optimization
Machine learning
Forecasting optimization solutions
Blood supply chain
Transshipment

ABSTRACT

Machine learning (ML) has attracted recent attention in solving constrained optimization problems to reduce computational time. In this article, we employ multi-output ML models including light gradient-boosting machine (LGBM) and multilayer perceptron (MLP) to predict the solution to a constrained optimization problem, explore the impacts of selecting different loss functions, and evaluate their performance by looking at the utility of predicted decisions, i.e., the associated cost components and the feasibility of predicted solutions We implement our approach in a blood supply chain where hospitals collaborate on transshipment to meet demand. We use demand distributions fitted to real data to evaluate the performance of the predictive models by analysis of the utility of forecasts, including total cost, inventory holding, transshipment, outdated unit, and shortage costs associated with predicted decisions. The results of our case study show that an LGBM model with the mean absolute deviation loss function provides solutions with only 2% higher total cost than a stochastic optimization model. Compared to an empirical policy, it could reduce transshipment and shortage costs by 23% and 6%, respectively. Therefore, altering the loss function of ML models can provide appropriate solutions to optimization problems, and various loss functions should be probed accordingly.

1. Introduction

Many organizations face complex daily operations decisions that can be formalized and informed by analytical techniques such as predictive and prescriptive models. Implementing these models in practice is mainly motivated by profit or social benefits and the effectiveness of these techniques in making informed decisions. However, some organizations, especially smaller ones, may be unable to leverage these models for several reasons, such as operational constraints, lack of expertise, time constraints, budget limitations, and limited infrastructure (Rostami-Tabar et al., 2022). More specifically, decision-making with constrained optimization models imposes some challenges in modeling and solving the problems.

For example, modeling and formulation of these decision problems cannot be automated. Further, the modeling process is often labor-intensive, requires a team effort of trained experts, and is deemed more of an art. Once these models are formulated, they can be implemented to assist with operational decisions after obtaining optimal solutions. However, depending on the type of problem at hand, they require different techniques which may be computationally expensive; for example, solving mixed-integer problems with many variables and complex feasible regions is known to be intractable in

* Corresponding author.
E-mail addresses: m.abolghasemi@uq.edu.au (M. Abolghasemi), babak.abbasi@rmit.edu.au (B. Abbasi), zahra.h@unimelb.edu.au (Z. HosseiniFard).

https://doi.org/10.1016/j.ijforecast.2023.05.004
0169-2070/

many settings (Bertsimas, King, & Mazumder, 2016). An example of such a problem is in healthcare organizations and hospitals, which must regularly make operational decisions to manage their operations. Hospitals often operate within a network where they transship blood units between hospitals to manage the blood inventory system, thereby reducing blood wastage. Since blood is a perishable and rare product, any improvement will likely have a significant socio-economic impact (Rostami-Tabar et al., 2022). In such a network, hospitals must decide how much blood to order from the central blood bank and how much blood to transship from other hospitals in their network under uncertain demand. In many cases, solutions to these problems are needed almost immediately. However, optimal solutions might only be available using commercial optimization solvers, which can take a considerable time for a large-scale problem (Abbasi, Babaei, Hosseinifard, Smith-Miles, & Dehghani, 2020). Furthermore, small hospitals may lack the technical expertise to implement and maintain the advanced big data analytics and the optimization models developed to optimize their supply chains, despite their potential benefits (Dahl, Milne, & Peltier, 2021; Kraus, Schiavone, Pluzhnikova, & Invernizzi, 2021). Due to these barriers, implementing optimization techniques in small hospitals with limited resources remains challenging. Machine learning (ML) techniques can provide an alternative way to model such problems with considerably less time and fewer resources.

A supervised ML technique can predict the outputs associated with new inputs through learning from input–output mappings. ML models are widely applied in various areas, including automation of operations in the supply chain, such as warehousing, forecasting demand, and inventory management. In the operations research literature, there has been limited work on using ML methods to predict the optimal solutions to constrained optimization problems, and there is no rigorous method or framework for modeling and solving these problems using ML (Abbasi et al., 2020; Bengio, Lodi, & Prouvost, 2021; Larsen et al., 2022). An approach commonly used in supervised learning is to provide a training dataset that uses the inputs and outputs of the optimization model to train the supervised ML model. For example, in the case of the blood supply chain, the inputs can be the blood inventory levels in each hospital, and the outputs can be the optimal decisions for the number of new orders and transshipment between hospitals in a typical network of hospitals. The model can be trained offline and used to generate solutions online. Such predictions can be made very fast in real-time. There are standard open-source programs with ample ML packages that can be used for training these models and making them more affordable for smaller organizations with limited resources. However, as *there is no free lunch*, these benefits do not come with a guarantee of obtaining optimal solutions or even feasibility. Therefore, it is necessary to check the feasibility and quality of solutions to ensure that the predicted solutions provide near-optimal results and satisfy the constraints, i.e., they do not violate any physical or engineering constraints.

This paper focuses on the impact of loss functions in ML models when they are used to predict the solutions to a complex transshipment problem in the blood supply chain. We consider various loss functions, including mean absolute error (MAE), mean squared error (MSE), and Huber loss, in optimizing the performance of the predictive ML models for effective learning from data. We also consider the feasibility of solutions to ensure the inventory constraints are met. By transferring blood among hospitals, hospitals can reduce wastage without incurring large costs on the supply chain. The goal is to reduce the total costs of the supply chain while ensuring an acceptable service level, i.e., meeting the demands of hospitals. Therefore, in evaluating our proposed models, we measured their performance by analyzing the total costs of the supply chain as well as individual cost components such as ordering, transshipment, outdated unit, and shortage costs.

1.1. Contributions

This study builds on the findings of Abbasi et al. (2020), who used ML models to predict the solution to a blood transshipment problem, also considered in this study. We have extended their work in four main directions:

- Multi-output regression: We aim to improve the performance of the predictive models by developing multi-output ML models. We consider well-known ML algorithms commonly used in forecasting tasks, namely light gradient-boosting machine (LGBM), multilayer perceptron (MLP), support vector regression (SVR), and ridge regression in a multi-input multi-output fashion to predict the values of the decision variables.
- Altering loss functions: We further explore the impact of the loss function when predicting the solutions to optimization problems. We train the LGBM and MLP models with MSE, MAE, and Huber loss functions to examine the performance of the loss function. Second, we investigate the performance of these models by examining the feasibility of the predicted solutions (i.e., whether the predicted solutions violate any constraints in the optimization model). We show that the loss function can significantly impact the validity of the predicted solutions.
- Feasibility evaluation: Since the predictive ML models do not consider the constraint violations of the optimization problem, this has been verified after the results were obtained to ensure the predicted solutions meet the constraints. We then evaluate their efficiency by comparing the number of occasions on which the ML models produced infeasible solutions.
- Forecast utility: We also look at the forecast utility rather than the forecast accuracy to evaluate the performance of ML models in predicting the solutions to the optimization problem. Forecast accuracy is often used as a proxy to measure forecasts' effectiveness. However, it is hard to infer how forecast accuracy will be translated into the final decisions. Here, we directly measure forecast performance in terms of costs imposed on the supply chain. We evaluate the performance of the forecasts in the

first stage of the objective function of a two-stage stochastic optimization problem by measuring ordering, transshipment, outdated unit, and shortage costs.

1.2. Organization of the paper

The rest of the paper is organized as follows: Section 2 is a review of related literature. Section 3 describes the methodology used throughout this study and presents the proposed ML forecast algorithm. Data and experimental setup are explained in Section 4, while Section 4.4 discusses the theoretical and managerial implications of the findings. Finally, we conclude the paper, propose future research directions in Section 5 and provide further experimental results in the Appendix.

2. Literature review

This study is related to the literature on ML for optimization, blood supply chain, and blood transshipment. In this section, we provide a detailed literature review related to our methodology and then refer the reader to recent literature review papers conducted in the blood supply chain.

2.1. Machine learning in optimization

Applications of ML in offering solutions to combinatorial optimization problems can be generally divided into two streams: (i) those that use ML to mitigate the computational burden associated with the mathematical modeling of optimization problems (Abbasi et al., 2020; Larsen et al., 2022), and exploring how ML can accelerate optimization algorithms and solvers (Kruber, Lübbecke, & Parmentier, 2017; Lodi & Zarpellon, 2017; Vaclavik, Novak, Scha, & Hanzlek, 2018), and (ii) those that use ML to help solve optimization problems that are not mathematically well-defined (He, Daume, & Eisner, 2014; Khalil, Dai, Zhang, Dilkina, & Song, 2017). While the latter case builds a solution from scratch, the first approach is a supervised learning task, which is also the focus of this study. There is also another stream of research on using ML to approximate the performance of various solutions that otherwise would need to be estimated through (often time-consuming) simulation. For example, (Fischetti & Fraccaro, 2019) used linear regression, neural networks, and support vector machines to approximate the performance of various offshore wind farm layouts. In another example of inventory management, Theodorou, Spiliotis, and Assimakopoulos (2023) used LightGBM to predict the inventory cost elements where the training dataset, including performance and various inventory model's hyperparameters, was obtained using simulation.

A limited number of studies have used ML models to predict the solutions to optimization problems (Abolghasemi, 2022). It is unclear how ML predictive models can be developed to mimic the behavior of constrained optimization models, optimize various decisions, and obtain reliable and valid solutions. Larsen et al. (2022) studied using ML models to predict the solutions for second-stage decisions in a two-stage stochastic model. They used deep learning models to solve the two-stage model and showed that their proposed model could find close solutions to the lower bounds of an average approximation method significantly faster, i.e., within milliseconds. Abbasi et al. (2020) proposed predicting the solutions for the first stage of a two-stage stochastic optimization problem and the optimal value of decision variables. They used four ML models, including classification and regression tree, k-nearest neighbor, random forest, and MLP, to predict the tactical solutions and total costs associated with them. They found that MLP was the top-performing model, but it underperformed the mathematical model by around 14%. While they checked that the solutions met the constraints, they did not evaluate their trained models' effectiveness in meeting them without manual intervention. Some studies have attempted to capture constraints using an automated approach, such as the Lagrangian method (Fioretto, Hentenryck, et al., 2020; Fioretto, Van Hentenryck, et al., 2020). Fioretto, Hentenryck, et al. (2020) developed a deep learning model and used the Lagrangian dual method to meet the constraints. Their empirical results on energy data were shown to be promising. Chatzos, Fioretto, Mak, and Van Hentenryck (2020) also used deep learning methods with Lagrangian duality and achieved solutions that were within 0.01% of the optimal solutions in only a few milliseconds, indicating the strength and feasibility of their proposed solutions in the context of power flow. A detailed literature review on the intersection of ML and optimization problems is provided in Bengio et al. (2021). Abolghasemi (2022) also indicates that using ML to predict the solutions to constrained optimization problems is an area that has not been extensively explored.

In the context of using ML to find solutions to optimization problems, the training data is often comprised of instances solved offline via constrained optimization solvers. This includes the parameters as inputs and the values of decision variables as outputs of the training data at each time stamp. The test set contains the parameters of the optimization model, which may not be known in advance but will be known progressively. Therefore, the temporal element in data is important, and models should consider it. To predict solutions to a blood supply chain optimization problem, Abbasi et al. (2020) used the solutions of two-stage stochastic models as the training dataset in a supervised ML framework. Specifically, hospitals' inventories on each day were the inputs, and decisions on the number of orders as well as transshipment decisions were considered as the output variables. They evaluated the performance and practical utility of most common ML models by simulating the blood supply chain model over 18,500 days and found some promising results. However, they did not empirically investigate the performance of different loss functions in learning and predicting decision variables' values and associated costs.

With the advancement of ML models, there has been a growing interest in developing customized loss functions for various predictive modeling tasks (Lin, Goyal, Girshick, He, & Dollár, 2017). However, estimating the loss function for a complicated optimization model may be difficult when using ML to predict the solutions. There is no unique loss function that can be used in ML models to solve different constrained optimization problems; rather, they must be estimated for the individual problem at hand (Bengio et al., 2021). Some methods may be more suitable for a particular class of problems, and forecasters need to determine an appropriate model architecture for their problem. While there are some general guidelines on choosing the appropriate type of model and loss function, researchers are more reliant on experiments to find their desirable architecture (Hewamalage, Ackermann, & Bergmeir, 2022; Makridakis, Spiliotis, & Assimakopoulos, 2018). It is not evident how the choice of loss function will impact the feasibility and quality of the optimization model's solutions. We investigate a case in the blood supply chain to shed light on this problem. We look at popular loss functions and measure their efficiency in terms of the quality of the final solutions as well as their ability to meet the inventory constraints of the optimization problem.

Mandi, Bucarey, Tchomba, and Guns (2022) used decision loss instead of estimating a loss function for their optimization model. The advantage of using decision loss is that it avoids the issues of complexity and non-differentiable optimization functions. Furthermore, using decision loss is closely tied to the utility of forecasts, i.e., by using decision loss, the utility of forecasts is directly introduced into the learning model. The utility of forecasts is an important phenomenon, particularly in supply chain forecasting (Syntetos, Babai, Boylan, Kolassa, & Nikolopoulos, 2016). The common approach to evaluating the performance of forecasting tasks is to compare the accuracy of the generated forecasts against their actual values. However, statistical measures do not necessarily reflect their actual performance in business forecasting applications. In other words, managers are often interested in the profit or loss resulting from their forecasting models, not the accuracy of the models (Armstrong, 2001).

While popular loss functions such as MAE (mean absolute error) and MAPE (absolute percentage error) might be relevant for measuring accuracy in some problems, they are criticized for their bias and irrelevancy in various problems (Boylan et al., 2006). Therefore, utility measures such as total inventory and production costs have been put forward as a more suitable way of measuring the performance of forecasting models (Ali, Boylan, & Syntetos, 2012). The utility of forecasts has been minimally considered, with most researchers looking at the utility of forecasts only on inventory costs (Syntetos & Boylan, 2006). In this study, we take a holistic approach to monitoring the utility of forecasts in the selected blood supply chain case study by examining the ordering, transportation, shortage, outdated units, and holding costs imposed by the generated forecasts. We build various multi-input multi-output ML models, train them with different loss functions, and evaluate their performance in solving the optimization problem by looking at the utility of the forecasts. We also measure their efficacy by the number of times that they violate the inventory constraints.

2.2. Blood supply chain

The blood supply chain (BSC) has been an active area of research for decades, as improving the blood supply chain can improve blood availability and ultimately save lives. This has led to many challenging optimization problems, such as blood transshipment. For a broad literature review on the blood supply chain, we refer readers to Belien and Force (2012), Osorio, Brailsford, and Smith (2015), Pirabán, Guerrero, and Labadie (2019), Torrado and Barbosa-Póvoa (2022) and Williams, Harper, and Gartner (2020). They discuss the studies on the design of a sustainable BSC, blood inventory management, collection, allocation decisions, and issuing policies under stochastic demand and replenishment. Further, Meneses, Santos, and Barbosa-Póvoa (2022) reviewed existing optimization models used to tackle planning decisions at the collection, production, and distribution stages of a BSC. In this paper, we focus on hospitals' inventory management in the blood supply chain and investigate applications of advanced ML techniques for solving a large-scale stochastic optimization problem.

To improve the performance of the BSC, outdated rates, shortages, and the age of transfused items should be minimized (Abbasi & Hosseinifard, 2014). As Stanger, Wilding, Hartmann, Yates, and Cotton (2013) and Abbasi, Vakili, and Chesneau (2017) discuss, the transshipment of blood units within a network of hospitals can also reduce the outdated rate, improve flexibility in blood supply management and enhance the performance of the blood supply chain. While Stanger et al. (2013) suggested the implementation of lateral transshipment to achieve efficient blood supply chain operations, Hosseinifard and Abbasi (2018) investigated the effect of complete centralization in the second echelon of the BSC, i.e., hospitals. They focus on reducing shortage, wastage, and the total costs of each hospital, responding to their demand and that of neighboring hospitals. Various analytical models, such as a two-stage stochastic framework, stochastic integer programming, and stochastic dynamic programming, have been developed and utilized to obtain optimal solutions for BSC inventory management (Arani, Momenitabar, Ebrahimi, & Liu, 2021; Gunpina & Centeno, 2015; Zhou, Leung, & Pierskalla, 2011). Dehghani, Abbasi, and Oliveira (2021) analyzed the transshipment policy in a small network of four hospitals to prevent future shortages and mitigate wastage. They employed a two-stage stochastic optimization model, compared the costs associated with their optimized ordering and transshipment policies with a no-transshipment policy, and showed the superiority of their model. However, solving the blood transshipment stochastic optimization problem involves heavy computations and access to somewhat expensive commercial software tools. This research will build on Dehghani et al. (2021)'s work and contribute to predicting solutions from stochastic optimization in the context of blood transshipment. The following section provides more details on the model and the ML prediction algorithm.

3. Method

We aim to build ML models to learn the relationship between a set of input parameters and output decision variables in the first stage of a two-stage stochastic optimization problem. Interested readers can see Dehghani et al. (2021) for a two-stage stochastic optimization model for our case study on the blood transshipment problem.

In two-stage stochastic optimization problems, the number of second-stage decision variables increases by the number of scenarios and the length of the planning horizon, and often the problem becomes large very quickly in many real-world applications with a large number of scenarios and long planning horizons. However, the number of actionable decision variables in such operational decision-making (i.e., the first-stage decision variables) is limited. Hence, it seems viable to design a mechanism for learning to produce the values of the first-stage decision variables of a two-stage stochastic optimization problem.

For such a problem, we need to consider ML models with multiple inputs, i.e., the parameters of the optimization model, and multiple outputs, i.e., the decision variables of the optimization model. The inputs and outputs may differ from one problem to another. In our case study, we have 18,500 observations with 44 inputs and 136 outputs (the same data used in Abbasi et al. (2020) and obtained by solving the two-stage stochastic optimization model developed by Dehghani et al. (2021) using Gurobi solver). Since 18,500 days of data is a long history and may not be available in many blood supply chain networks, we only use 10% of the original data, i.e., 1850 daily observations, which is equivalent to roughly five years of data, for the modeling and analysis in the main body of this study, and we provide further results with 9250 days in Appendix B. The input variables include the inventory levels of 11 different blood units with different ages for each of the four hospitals, totaling 44 variables. The output variables correspond to the number of transshipment of different aged blood units between hospitals (132 variables) and the number of orders each hospital places to the central blood bank (4 variables), totaling 136 decision variables.

To train and test the performance of an ML model in solving a constrained optimization model, we need to create a large dataset. To solve a two-stage stochastic model, we typically provide the parameters of the objective function and the problem's constraints to a solver and obtain the optimal solutions for the decision variables. Therefore, the set of inputs and outputs in such a problem can be organized in a dataset and trained via an ML model. For example, we split the original dataset into train and test sets in our case study. We used the first 90% of the observations of data to train the ML models. We then use the trained ML models to predict the solutions for the test set on a rolling origin basis. We only predict the solutions for the next period in the test set. Once the predicted order and transshipment quantities with different ages are obtained, we compute the other parameters of the model, including transshipment between hospitals, and update their inventory. Then, we rolled the model one step ahead and predicted the results for the next period. We continued this process until the end of the test set. The steps for problems similar to our case study are summarized in Algorithm 1.

Algorithm 1: ML prediction algorithm

1: **Offline phase (Preparing the training dataset and building ML models)**
2: Formulate the optimization problem in hand with an appropriate mathematical model.
3: Solve the mathematical model with a numerical solver to find the optimal values for the decision variables.
4: Build a simulation model that mimics the problem behaviors to determine the values of other required decision variables.
5: Store the parameters and decision variables as a dataset.
6: **Online phase (using a rolling horizon with length n for performance evaluation)**
7: Split the dataset obtained in step 4 into training and test sets. The training set is the older x% of data and the test set is the more recent 1-x%.
8: Develop and train a multivariate ML model of choice with an appropriate loss function. The inputs are the parameters of the mathematical model, and the outputs are the decision variables obtained in Step 4.
9: **For** $h = 1$ *to* n
10: Predict the decision variables.
11: Check the validity of the predicted decision variables to ensure they meet the constraints.
12: Replace the values of the decision variables in the supply chain simulation model to obtain the values of other required variables in the supply chain.
13: Replace the obtained values of the decision variables in the objective function of the mathematical model to compute the objective function value.
14: **End For**

There is no unique ML model capable of forecasting all types of data more accurately than other models under all conditions (Abolghasemi, Beh, Tarr, & Gerlach, 2020). However, empirical studies suggest certain models may be more effective for certain types of data (Petropoulos, Makridakis, Assimakopoulos, & Nikolopoulos, 2014). Given that we have sparse data comprising the parameters of the optimization model as inputs and the optimal values of decision variables as outputs, we use ML models with regularization. We implement ridge regression and SVR, standard benchmarking models that are promising in many problems. We also implement the different architectures of two widely successful ML algorithms, MLP and LGBM, with various loss functions (Abolghasemi, Tarr, & Bergmeir, 2022; Drucker et al., 1997; Hoerl & Kennard, 1970; Ke et al., 2017).

Ridge regression is a regression model that deals with multicollinearity problems in linear regression (Hoerl & Kennard, 1970). Multicollinearity often occurs when the number of parameters is large, potentially, they are not independent. Ridge regression performs the $l2$ regularization technique with a penalty cost, λ, to estimate the model's parameters and avoid biased results. The ridge loss function is defined in Eq. (1):

$$L(\beta, \lambda) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 n + \lambda \sum_{j=1}^{p} \beta_j^2, \quad (1)$$

where y is the actual value of the target variable, $\hat{y}$ is the predicted value of the target variable, λ is the ridge estimator, n is the number of observations, and p is the number of variables. The first part of the loss function is the sum of the squared error loss function, and the second part is the ridge penalty. The ridge loss function guarantees it rests at the global minimum for a convex function by finding the β and λ values. Since the range of parameter values in the output vector changes significantly, the ridge regression penalizes large model weights, making it suitable for obtaining unbiased blood supply chain model results. This model has been successfully implemented in various forecasting applications (Exterkate, Groenen, Heij, & van Dijk, 2016). We implement this model in a multi-input multi-output fashion by using the scikit-learn library in Python and applying the grid search cross-validation technique to obtain the optimal value of λ.

SVR is a powerful supervised learning algorithm (Burges, 1998). SVR is different from an ordinary least square regression model in that SVR attempts to minimize the generalized error rather than minimizing the deviation of predicted values from the actual ones. SVR uses a kernel function to map the input data to higher dimensional space in a non-linear fashion. SVR finds a function in the space within a distance of ϵ and ϵ^* from its predicted values. Any violation of this distance is penalized by a constant penalty cost, c. For a given set of data points (x, y), SVR solves the following constrained optimization problem to estimate the parameters.

$$\min_{w,b,\xi,\xi^*} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i + C \sum_{i=1}^{n} \xi_i^* \quad (2)$$

$$y_i - w^T \phi(x_i) - b \leq \epsilon + \xi_i,$$
$$w^T \phi(x_i) + b - y_i \leq \epsilon + \xi_i^*,$$
$$\xi_i, \xi_i^* \geq 0, i = 1, \ldots, n.$$

SVR has been effectively applied in many forecasting applications, including supply chain forecasting problems (Abolghasemi, Beh, et al., 2020; Abolghasemi, Hurley, Eshragh, & Fahimnia, 2020; Levis & Papageorgiou, 2005). We chose the radial basis as the kernel function and optimized the cost of the constraint violation, C, using the grid search cross-validation technique.

LGBM implements gradient-boosted decision trees based on ensembling and uses a number of hyperparameters for training models and generating forecasts (Ke et al., 2017). LGBM has been implemented in various forecasting problems and has attracted many researchers' and practitioners' attention by winning a number of forecasting competitions (Abolghasemi & Esmaeilbeigi, 2021; Makridakis & Spiliotis, 2021). LGBM is a fast and powerful algorithm that can handle various features, making it appealing for large-scale problems with diverse input variables. We implemented this model in a multi-output fashion to predict the solutions to the constrained optimization problem. The LGBM algorithm benefits from a large number of hyperparameters to learn the process and project them to the gradient space. We set the boosting type to *gbdt* and changed the objective as required to *mae*, *mse* or *Huber*, and found the optimal values of *max-depth*, *learning-rate*, *num-leaves* with the grid search cross-validation technique.

Multilayer perceptron (MLP) is a feedforward neural network that uses an input layer, one or more hidden layers, and an output layer. The layers are fully connected, with each layer consisting of neurons mapped to outputs by some activation function to learn the behavior of data. MLPs are universal approximators and can approximate almost any non-linear behavior with appropriate architecture and configuration, i.e., the number of neurons and layers, learning rate, activation function, etc. However, they are prone to overfitting, and one must implement rigorous validation to avoid this issue. We used two hidden layers with seven neurons, applied *Relu* as the activation function, and used a grid search to find the optimal learning rate values. We used *hep-ml* and scikit-learn to implement these models in Python 3.2.

We used a *MultiOutputRegressor* wrapper to generate multiple outputs at once. While MLP (with MSE loss function) and Ridge in scikit-learn naively, support the multi-output models, and can be optimized globally for all outputs, SVR and LGBM are single-output models. They may not be globally optimal for all outputs. We trained several LGBM and MLP models with different loss functions, as explained in Section 3.1.[1]

3.1. Loss function

The loss function is an important part of learning in ML models. The true loss function for an optimization or forecast problem is often difficult to estimate as its distribution is unknown. There is no consensus in the literature on choosing the best loss function (Clements & Hendry, 1993, 1995), as it depends on the dataset, the problem at hand, and the decision maker's objectives. In academia, statistical loss functions are widely used for training and evaluating models' performances. However, in the real world, the loss function is measured in dollar terms. Nevertheless, reliability, robustness to outliers, and comprehensibility are desirable criteria for a good loss function. We employed mean squared error (MSE), mean absolute error (MAE), and Huber loss, three widely known loss functions. We empirically evaluated their performance in our case study blood supply chain problem. We first used the well-known and popular MSE as the loss function. MSE is widely used in regression problems and is the default loss function for various ML and statistical models. MSE tries to find the best parameter values by minimizing the average error across all observations. The MSE loss is calculated as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \quad (3)$$

[1] The code is publicly available at https://github.com/mahdiabolghasemi/ML4predicting_optimisation.

where y is the actual value of the target variable and $\hat{y}$ is the predicted value of the target variable.

For the second attempt, we used MAE as the loss function. MAE is another popular loss function that is frequently used in various settings. MAE loss is calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{4}$$

where y is the actual value of the target variable and $\hat{y}$ is the predicted value of the target variable. The loss function MAE optimizes the learning process by considering the median of the values. As such, it is an appropriate choice to avoid the large impact of outliers that may be imposed on the learning process. This contrasts with MSE, which optimizes the mean across all forecasts. MSE penalizes any violation with a large cost, making it a non-robust cost function, especially when there are outliers in the forecasts. While MAE is a robust estimator, it can be biased because the gradient is not dependent on the size of the error but only on the sign of the error. That is, if the error is negative, the gradient is -1, and when the error is positive, the gradient takes the value of +1. This can be problematic and cause convergence problems when the error is small. We also implemented Huber as another loss function. Huber loss is calculated as follows:

$$Huber_\delta(y,\hat{y}) = \begin{cases} \frac{(y-\hat{y})^2}{2}, & |y-\hat{y}| \le \delta \\ \delta * (|y-\hat{y}|) - \frac{\delta^2}{2}, & otherwise \end{cases} \tag{5}$$

where y is the actual value of the target variable and $\hat{y}$ is the predicted value of the target variable.

Essentially, Huber loss combines the MSE and MAE loss functions to overcome their drawbacks. The Huber loss function penalizes with MSE for loss values smaller than δ, which tries to minimize the average of the errors. For larger errors when loss values are greater than δ, the Huber loss function penalizes with a similar function to MAE. It uses the δ term as a regularizer to dampen the impact of the outliers. We use cross-validation technique to determine the optimal value of δ with different values for δ ranging from 1 to 10, with steps of 1.

Other loss functions, such as popular MAPE, can be considered. However, we do not use MAPE in this study as the mathematical properties of MAPE are not desirable for our problem. This is because we have parameters with the value of Zero, making MAPE an infinite number in some scenarios. Therefore, we skip MAPE in our analysis. Nevertheless, MAPE or other appropriate loss functions, including customized loss functions, can be considered alternative loss functions.

4. Data and experimental setup

4.1. Data and case study

The data are related to a case study comprising four hospitals and a central blood bank. We gathered blood data from four hospitals. For confidentiality reasons, we were not allowed to use the same data. Therefore, we estimated the demand distribution and used fitted zero-inflated negative binomial demand distributions on real demand data from four hospitals (Dehghani et al., 2021). The zero-inflated negative binomial distribution combines negative binomial and logit distributions. For such a demand, the distribution outputs take non-negative integer values. This ought to be a realistic choice for our empirical blood demand as demand for blood units can be zero, for example, over the weekends or take a positive integer on weekdays. The zero-inflated negative binomial distribution has three parameters: the number of trials (r) and the probability of success in each trial (p), which correspond to the negative binomial part of the distribution, and the inflated probability of zero (π), which corresponds to the logit part (Doyle, 2009). The parameters of the demand distribution for four hospitals are ZINB($\pi = 0.6$, $r = 4$, $p = 0.6$), ZINB($\pi = 0.6$, $r = 3$, $p = 0.57$), ZINB($\pi = 0.25$, $r = 15$, $p = 0.57$) and ZINB($\pi = 0.25$, $r = 15$, $p = 0.48$).

In such a network, hospitals can satisfy their demand by ordering fresh blood units from the central blood bank or transshipping blood units from other hospitals. These decisions depend on the demand and availability of blood in other hospitals. They have to make decisions at the beginning of each day when the demand for the day is still unknown. If they have excess blood units, a holding cost is incurred, and if they have blood units that are older than 11 days, they have to discard them with the cost of wastage. Dehghani et al. (2021) solved the developed two-stage stochastic optimization using the Gurobi solver, and we used their solutions to develop the ML models in this study. The objective function of the optimization problem is comprised of holding costs, shortage costs, outdated unit costs, and transshipment costs. The value of the holding cost, emergency order cost, outdated unit cost, order cost, and transshipment cost were set at 1, 14, 11, 1, and 1.5 monetary units per unit per period, respectively.

4.2. Experimental setup

The dataset for training our ML models included inventory levels for four hospitals and the optimal values of orders for the entire course of the simulation. In total, we used 1850 observations. This dataset is provided from running a two-stage stochastic model to optimality over 1850 days to provide robust data for training and testing our models. The input variables of the ML models represent the inventory level of four hospitals. The output variables correspond to the orders of each hospital from the blood center (four decision variables) and transshipment quantities between one hospital and others with a maximum shelf life of 11 days (132 decision variables = 4 hospitals × 11 blood units with different ages × 3 transshipment to three other hospitals). More specifically, the inputs of the ML models are the number of units with age m ($1 \le m \le 11$) at each hospital, i.e., the inventory level of each hospital, and the outputs are the orders of each hospital from the central blood bank and the transshipment of units with age m ($1 \le m \le 11$) from hospital i to j.

We looked at the forecasts' utility to evaluate the ML models' performance. That is, instead of measuring forecast accuracy by statistical metrics such as MAE or MSE, we measured their performance by replacing the predicted values of the decision variables in the first stage of the objective function to compute various incurred costs including inventory costs, ordering costs, transshipment costs, outdated unit costs, and shortage costs, and the total costs of the supply chain. The cost functions of the ML models are MAE, MSE, and Huber, as discussed in Section 3.1. This means the ML models assume that a more accurate forecast of decisions will lead to lower costs, although this is not guaranteed. So, we report the associated costs of the predicted decisions as our evaluation metric. Although we do not report the statistical accuracy metrics, as they do not represent meaningful information for decision-makers in our blood supply chain case study, we state that there is no guarantee that the best-performing models in our setting generate more statistically accurate forecasts either. We also implemented a customized loss function where the gradient and Hessian of the objective function were used for learning in ML models. Since the objective function of the optimization model is linear, the gradient is a constant number, and the Hessian is zero. This did not lead to good results, so we do not report them.

Once the optimal values of decision variables for orders and transshipment were predicted, we needed to replace them in the supply chain simulation process, which mimicked the process in practice, to determine the values of other decision variables and their associated costs. We summarize this process as follows.

- (i) We initiated the problem by providing the available inventory of blood units for each hospital.
- (ii) We used the ML models to predict the order quantities from the central bank and the transshipment quantities between hospitals. If the predicted value for transshipment was larger than the available inventory for a hospital, then the whole inventory would be transshipped. If a negative value was predicted, we replaced it with zero. Once these values were determined, we calculated the orders and transshipment costs.
- (iii) Hospitals update their inventory level after transshipping the blood units.
- (iv) Hospitals realize and fulfill their demand using their available inventory. Since the demand was known, the corresponding shortage cost was calculated, if any.
- (v) The transshipped orders are received at the end of the day, and hospitals update their inventory level accordingly. At this stage, the outdated unit costs and holding costs were calculated.
- (vi) The total cost was calculated by summing up the aforementioned costs, i.e., ordering, transportation, shortage, outdated unit, and holding costs.

4.3. Empirical results

We implemented the ML methods as described in Section 3 and looked at the forecast utility by evaluating the first stage of the objective function of the blood supply chain model (presented in Dehghani et al., 2021). We first predicted the values of the decision variables daily (the number of blood units to be ordered from the central bank and transshipment between different hospitals). Then, we replaced the obtained values in the simulated blood supply chain model to obtain the transportation, holding, outdated unit, transshipment, and shortage costs over 1850 simulation days. We benchmarked our models against SVR and ridge models, as two common ML models with their loss functions, the two-stage (TS) stochastic optimization model and current policy. Note that we do not use more simplistic models, such as Naive, as a benchmarking method because we have multiple outputs that need to be predicted, and these values are interrelated, making the Naive model ineffective in our case. The current policy of the hospitals employs an order-up-to policy where decision-makers observe the inventory level daily. If it is below a certain threshold, i.e., four times the average daily demand, they place an order to meet the threshold demand. The required blood units with a residual life of less than a certain value (six days) will be transshipped from the smaller hospitals (hospitals 1 and 2) to larger ones (hospitals 3 and 4).

Table 1 displays the performance of the ML methods considered in this study as well as the TS model and current policy. The results are reported for all involved costs, including holding, transshipment, ordering, shortage, and outdated unit costs.

The results based on the total cost indicate that, on average, among the ML models, LGBM trained with MAE loss function (LGBM-MAE) generates the lowest total costs in the supply chain. The incurred costs with LGBM-MAE predictions are very close to the optimal TS model, with the LGBM-MAE model performing only 2.6% sub-optimally. Considering the other costs, we can see that the LGBM-MAE model performs well and outperforms the TS model in reducing shortage and outdated unit costs. This is a great advantage for the model since blood shortages can have a dramatic impact in case of an emergency.

The current policy, on the other hand, has the smallest shortage costs at the expense of higher holding costs. The current policy has significantly higher holding costs than all other models, contributing to a larger total cost. Observe that the current policy also has the largest transshipment cost. This indicates that the current policy is a conservative approach that tries to minimize the shortage and outdated unit costs by excess transshipment of blood units between hospitals and imposing additional costs on the supply chain. Our results with LGBM-MSE and MLP-MSE show similar outcomes where higher holding costs have resulted in lower shortage costs, although they may not be optimal. In terms of outdated unit costs, the TS model has the best performance, followed by the LGBM-MAE model. The SVR and ridge models have the lowest transportation costs. This does not make them superior to other models because the lower transportation costs have contributed to relatively higher shortages and outdated unit costs. As is evident from the results, both the ridge and SVR models have a significantly higher shortage and outdated unit cost than others, except the MLP-MSE and LGBM-MSE models, which have slightly higher outdated

Table 1
Forecasting performance of ML methods in terms of average imposed costs over 1850 days (Best two performers in each cost are in bold.).

Models	Holding	Transshipment	Outdated	Ordering	Shortage	Total
LGBM-MSE	49.89	1.02	2.81	22.41	**4.75**	80.90
LGBM-MAE	**45.17**	1.70	**1.90**	22.52	5.12	**76.41**
LGBM-Huber	47.20	1.40	1.90	22.40	5.18	78.07
MLP-MSE	50.29	1.24	2.62	22.41	5.60	82.18
MLP-MAE	45.71	1.48	2.18	**22.07**	5.64	77.09
MLP-Huber	45.49	1.52	2.27	22.17	7.07	78.54
Ridge	45.25	**0.46**	2.32	22.42	7.26	77.71
SVR	45.22	**0.59**	2.55	**22.34**	7.47	78.17
TS model	**42.73**	2.43	**1.59**	22.37	5.33	**74.51**
Current policy	87.61	3.61	2.51	22.61	**3.12**	119.49

unit costs. All newly proposed ML models outperform the MLP-MSE model, the best-performing model in the previous study on similar blood data sets (Abbasi et al., 2020).

Comparing the performance of the loss function in the MLP and LGBM models reveals that both models with the MSE loss function have lower transshipment costs than their counterparts with the MAE and Huber loss functions, which in turn has resulted in higher holding costs and outdated unit costs. Large holding and outdated unit costs verify that, on the one hand, the models with the MSE loss function have not been effective in predicting the decision variables, thus misspecifying the required transshipment between hospitals and leading to higher holding and outdated unit costs. Models with the Huber loss function often resulted somewhere between the MAE and MSE loss functions and were not the top performer in any of the considered cost items. This indicated that, while the MAE and MSE loss functions can be used for optimizing various decisions, Huber plays in the middle ground, considering an average of both without sacrificing too much in the quality of one decision. Also, we generally observe that the LGBM model outperforms the MLP counterparts in most of the measured metrics.

In summary, while models with MAE loss functions outperform their counterparts with MSE loss functions by about 5%, the performance of models with the Huber loss function is somewhere between the MAE and MSE loss functions. The associated costs for the predicted results by the LGBM-MAE model are only 2.6% more than the optimal TS model. This low cost is evident in the total cost. LGBM-MAE predictions have consistently resulted in low costs in holding, transshipment, outdated units, ordering, and shortages, making it a robust model. The low cost of the LGBM-MAE model indicates that this model has effectively managed to predict demands and transship the blood units between hospitals while minimizing the holding, outdated units, and shortage costs.

The results show that the loss function can play a pivotal role in determining the performance of the models for various decisions that decision-makers may wish to optimize. This is evident from the utility of forecasts, which are translated to monetary values. However, different loss functions may perform better for specific objectives. For example, LGBM-MSE has outperformed other models in terms of transshipment costs. While this may not be the optimal policy for the blood supply chain case, one can choose an appropriate model to minimize this desirable objective. We assert that it is imperative to choose the appropriate loss function according to the model's parameters, the problem at hand, and the decision maker's interests. Optimizing one decision may not optimize the other decisions, and one needs to consider one's objective in choosing appropriate models.

To evaluate the performance of the models across all observations, we investigated the distribution of the bloods inventory for the investigated ML models at each hospital. The results, presented for each hospital and each ML model, are depicted in Figs. 1 where box-plots are used to display the minimum, 1st quantile, median, 3rd quantile, and maximum values of the costs, as well as any possible outliers. For readability purposes, we only show the results for LGBM-MAE and TS models as two top-performing models. Additional graphs are provided in Appendix A for all models and hospitals. As we can see in Fig. 1, the inventory levels for hospitals differ greatly for blood units of different ages. All hospitals have lower inventory levels for younger blood units, while TS generates a lower inventory level for older blood units.

As discussed before, when we use unconstrained ML models to predict the solutions to constrained optimization problems, we should monitor the solutions and observe whether the predicted solutions by ML models meet the problem's constraints. Table 2 shows how many times each of the presented ML models violated this constraint. That is, the ML model predicted a solution for the number of orders that was higher than the available inventory. While in the ML models, we force the predicted solutions to be smaller than or equal to the inventory levels of hospitals, in conventional constrained optimization models, these constraints are considered along with other constraints and parameters of the model to search the feasible space and find the optimal solutions.

As can be seen in Table A.2, the SVR and LGBM-MAE models have the lowest number of constraint violations, with only 9 and 11 violations occurring in 1850 days, respectively. That is 0.5% of observations. We observe that the solutions generated by an MSE loss function violated the constraints more frequently than other loss functions, making it less attractive for learning the decisions. The number of violations by the Huber loss function is between those caused by MAE and MSE. Lastly, we can see that MLP performed poorly compared to other investigated models, violating the constraints 89, 69, and

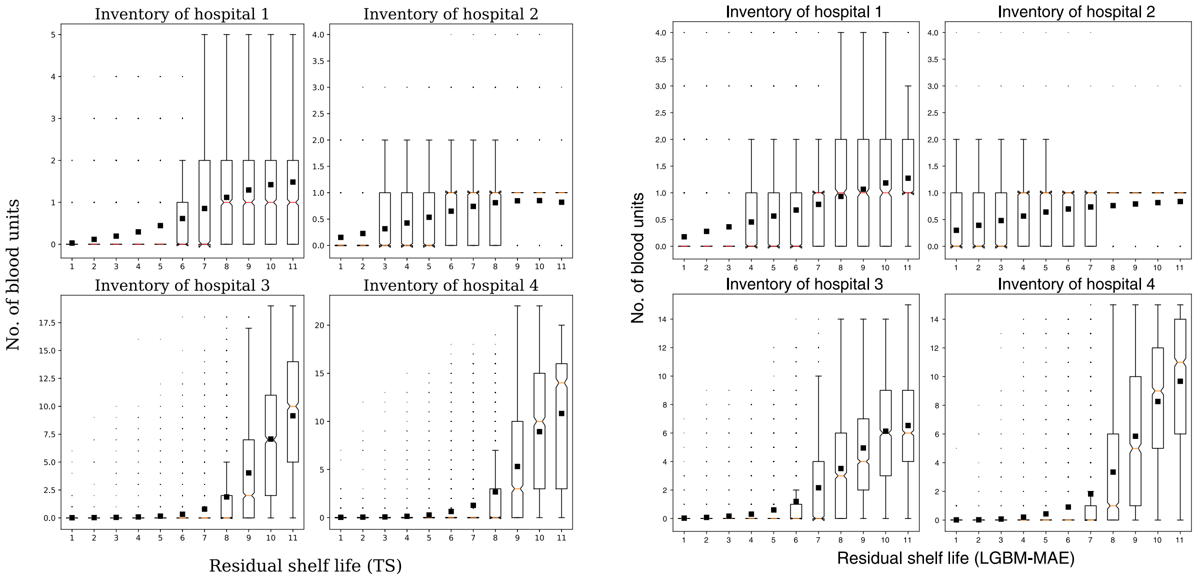

Fig. 1. The distribution of the inventory level of blood units of different ages at each hospital according to the TS and LGBM-MAE models. The inventory level is used to predict transshipment decisions and is vitally important in determining the performance and cost of the supply chain.

Table 2
The number of times the solutions predicted by the ML models violated a constraint of the inventory level.

Ridge	SVR	LGBM-MSE	LGBM-MAE	LGBM-Huber	MLP-MSE	MLP-MAE	MLP-Huber
54	9	44	11	19	89	69	70

Table 3
Computational time required to solve the problem in seconds.

TS	Ridge	SVR	LGBM-MSE	LGBM-MAE	LGBM-Huber	MLP-MSE	MLP-MAE	MLP-Huber
1200	0.17	0.14	0.14	0.14	0.14	0.21	0.21	0.21

70 times for ML-MSE, MLP-MAE, and MLP-Huber models, respectively.

As mentioned before, one important advantage of using ML models is their ability to generate fast and reliable forecasts that significantly save computational costs. Table 3 shows the computational time required to solve the problem at each time stamp in our study. We observe that LGBM models are significantly faster than the exact mathematical model while generating competitive results. Other ML models are also computationally fast, at least several thousand times faster than the TS model.

4.4. Discussion

In this section, we present some theoretical implications and managerial insights.

4.5. Theoretical implications

In the two-stage stochastic optimization model, several constraints are imposed by the case study problem for the inventory levels of hospitals and, therefore, must be met. We used ML models to predict the solutions to a constrained optimization problem where a set of inputs (inventory levels of hospitals) were given, and the most probable outputs (transshipment orders and orders to the central blood bank) were predicted. The predicted transshipment may be larger than the available hospital inventory, thus violating constraints in the optimization model.

In Table 2, we observed that the LGBM-MAE model could mimic the solutions to constrained optimization problems more effectively while significantly committing to the constraints. In this study, we manually enforced the solutions by hard coding to meet the constraints and ensure the feasibility of the solutions. We kept a record of violations and corrected them immediately to ensure the validity of the results. Our modeling with different loss functions proved effective, with only a few violations occurring in the unconstrained ML models. However, a more automated method to consider and learn the constraints in the training phase of the ML model would be useful in theory and practice. Although learning the feasibility of solutions in the ML model may not be trivial, some studies have used Lagrangian duality to embed the constraints in the objective function of the model (Fioretto, Mak, & Van Hentenryck, 2020). However, these methods are soft, i.e., there is no guarantee that violations will not

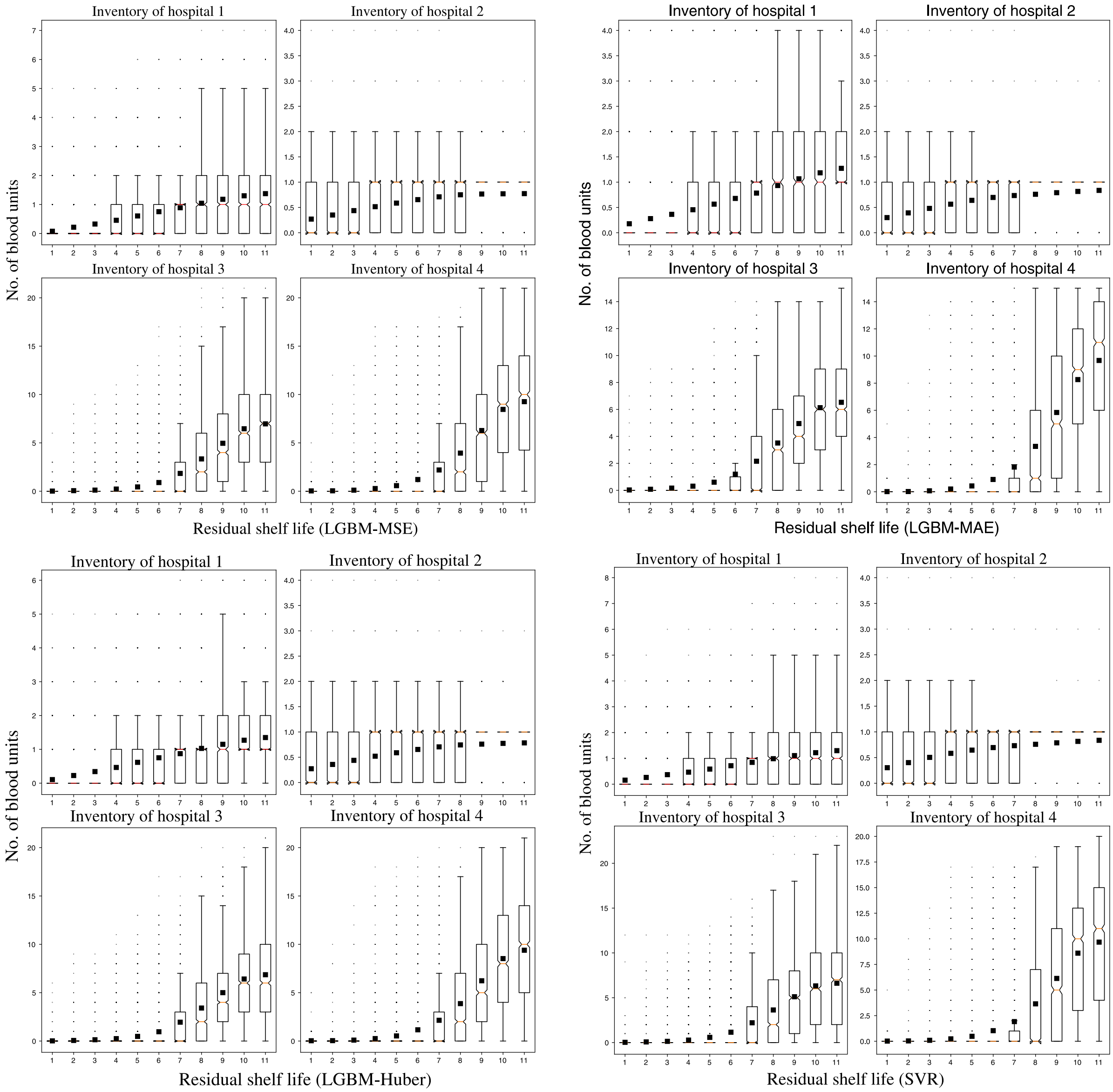

Fig. A.1. The distribution of the inventory level of blood units of different ages at each hospital according to the LGBM and SVR models. The inventory level is used to predict transshipment decisions and is vitally important in determining the performance and cost of the supply chain.

occur, and committing to the constraints may come at the expense of objective functions.

The computational speed of ML models is one of the main advantages of using them for estimating the solutions to optimization problems, making them effective tools for online decision-making where instant decisions are required. We note that there is no significant difference in the computational speed of the ML models in the presence or absence of additional downstream decision information.

4.6. Managerial implications

The empirical results show that we can use ML models to mimic the solutions to optimization problems. The ML models can be fast and reliable as they generate solutions in only seconds and largely abide by the constraints that may be present in the model. These models can be automated and implemented in open-source programming languages without requiring investment in software. This makes them appealing to use in practice. The framework proposed in this study can serve as a benchmark to implement these models. However, these models should be used with care. Firstly, any changes in the parameters may change the optimal solutions. Therefore, the optimization model must be rerun and the ML model trained accordingly. Secondly, if the optimization models are not solved to optimality, e.g., if the mixed integer programming models are solved to a certain optimality gap, then

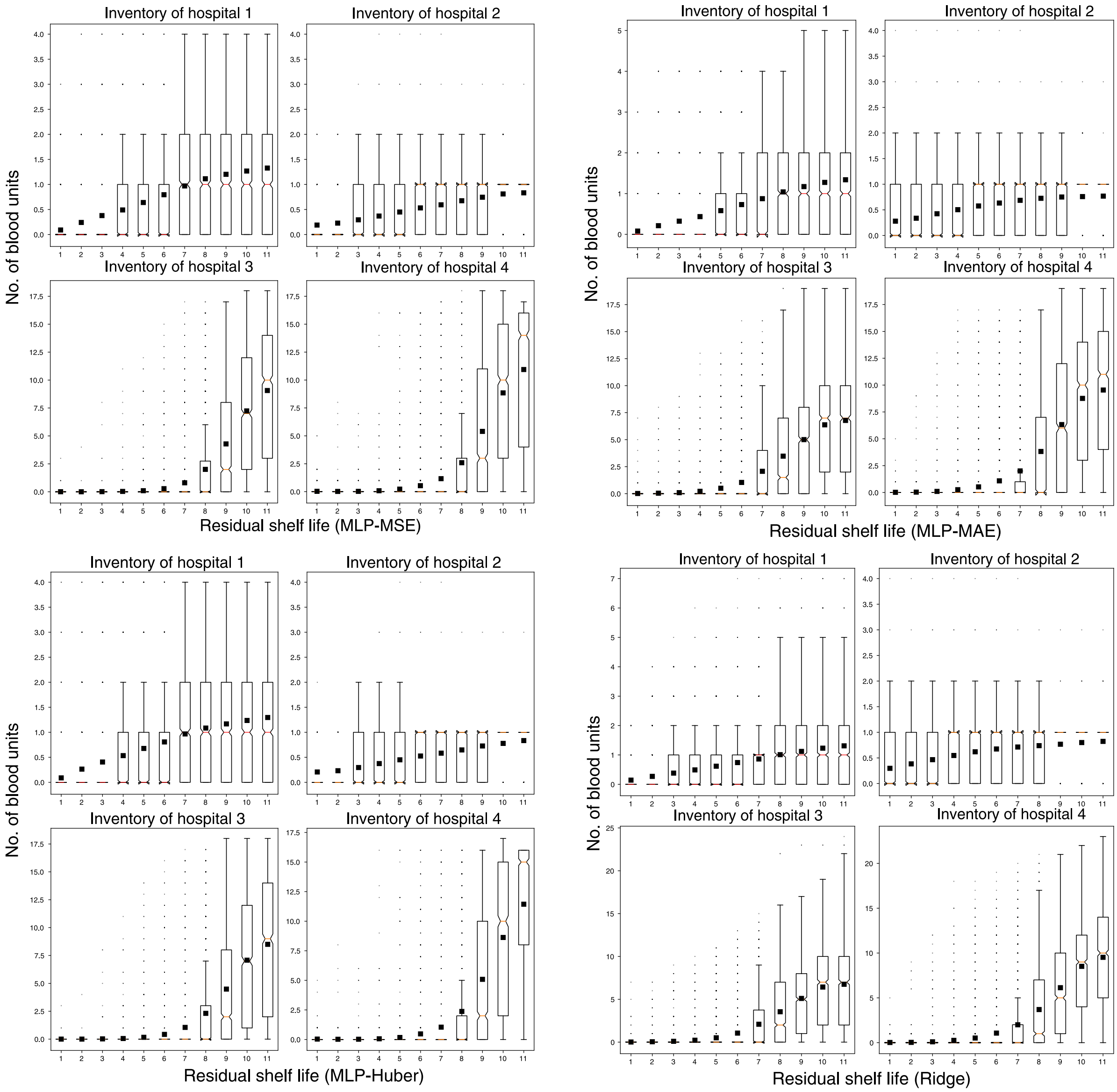

Fig. A.2. The distribution of the inventory level of blood units of different ages at each hospital according to the MLP and ridge models. The inventory level is used to predict transshipment decisions and is vitally important in determining the performance and cost of the supply chain.

the training data for ML models are not optimal and may be misleading. ML models generally cannot outperform the optimization model unless to a marginal level. It is important to solve the optimization model as accurately as possible. Thirdly, as discussed in this study, the ML models do not have a notion of constraints, and including sophisticated constraints in ML models is not trivial. Instead, the constraints should be hard coded in a heuristic manner in the algorithms to ensure consistency with the constraints in place.

Table A.2 shows the performance of the models in terms of constraint violations. As we can see, the SVR and LGBM-MAE models have generated more reliable solutions and only violated the inventory level 14 and 15 times (among 9250 days), respectively.

5. Conclusion and recommendations

This paper introduced a framework for using multi-output ML models to predict the solutions to a constrained optimization problem in a blood inventory management system. The motivation was to develop a fast model that is not computationally expensive and can generate reliable solutions in real time for instant decision-making. We explored the ML models' performance by investigating the forecasts' utility and examining different costs associated with the generated solutions. Instead of looking at traditional forecasting accuracy metrics, we computed the holding, transshipment, outdated unit, order, and shortage costs associated with the predicted decision variables to evaluate the performance of the ML models. Our results indicate that we can use

Table A.1
Forecasting performance of ML methods in terms of average imposed costs over 9250 days. (Best two performers in each cost are in bold).

Models	Holding	Transhipment	Outdate	Ordering	Shortage	Total
LGBM-MSE	48.38	1.04	2.60	20.87	7.76	80.05
LGBM-MAE	**45.34**	2.00	1.82	22.41	**5.26**	76.86
LGBM-Huber	46.22	1.44	**1.83**	22.44	5.28	79.40
MLP-MSE	47.83	1.41	1.98	22.83	6.74	80.81
MLP-MAE	45.51	1.48	2.18	22.37	5.80	77.36
MLP-Huber	46.29	1.16	2.51	23.53	7.18	80.68
Ridge	45.36	**0.85**	2.39	**22.33**	7.29	78.25
SVR	45.54	**0.59**	2.53	**22.32**	7.64	78.63
TS model	**42.73**	2.43	**1.59**	22.37	5.33	**74.51**
Current policy	87.61	3.61	2.51	22.61	**3.12**	119.49

Table A.2
The number of times that predicted decision of the ML models violated a constraint of the inventory level.

Ridge	SVR	LGBM-MSE	LGBM-MAE	LGBM-Huber	MLP-MSE	MLP-MAE	MLP-Huber
112	14	25	15	32	235	202	219

ML models to forecast the optimal values of the decision variables with up to 98% similarity to the optimal solution while committing to constraints over 99% of the time. We investigated the role of the loss function in predicting the solutions to optimization problems and various associated decisions. To do so, we trained LGBM and MLP models with the MAE, MSE, and Huber loss functions and showed that overall using the MAE loss function leads to a better performance than using the MSE and Huber loss functions for the problem in hand. However, different loss functions may optimize different decisions, and one should choose an appropriate loss function to optimize one's desired objective.

In this study, while we focused on the various supply chain costs to measure the models' performance, we optimized our algorithm with statistical loss functions. However, one could consider the optimization model's objectives and optimize the ML model's learning process accordingly. One way to extend this research would be to train the ML models with a customized loss function. This customized loss function should be estimated by considering the objective function and constraints of the problem at hand. This may not always be feasible if the objective function is not differentiable or may be difficult to estimate for some problems, given that the existing constraints can add extra complexity to the objective function that should be embedded as a cost function in the ML models. Another natural extension to our study would be to develop probabilistic forecasting models to estimate the solutions to optimization problems. Probabilistic models could provide a range of feasible solutions for each decision variable. The solutions could be accepted according to the level of risk one is willing to take.

Other hospitals with blood supply chains and a similar setting could adopt the proposed framework in this study to generate their required solutions. We fitted our model on real data and simulated the supply chain process in a network of four hospitals. Our results using ML models to predict blood transshipment decisions were promising based on the simulations and the case study. Empirical results and insights by other researchers on using open-source and fast ML models to solve constrained optimization problems can enormously benefit practitioners and the operations research societies. There is a need for research to understand better the impact of the loss function on the accuracy of models trained to mimic the behavior of constrained optimization models. Furthermore, there is a growing area of research called "predict to optimize" that tries to combine prediction and optimization (Abolghasemi & Bean, 2022; Elmachtoub, Liang, & McNellis, 2020). As the parameters of an optimization model are traditionally predicted first and then used in an optimization model to obtain the optimal solutions, we suggest exploring how ML can be used to integrate prediction and optimization in operational decision-making, particularly where the optimization problem must be solved sequentially. As an example, predicting the final solutions can be performed based on initial (historical) data rather than predicting the parameters of the optimization model in the first place.

Declaration of competing interest

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

Acknowledgment

The authors would like to thank the associate editor, guest editors, and reviewers for providing constructive comments on an earlier version of this article.

Appendix A

In order to evaluate the performance of the models across 1850 days of observations, we investigated the distribution of the bloods inventory for the investigated ML models at each hospital. The results, presented for

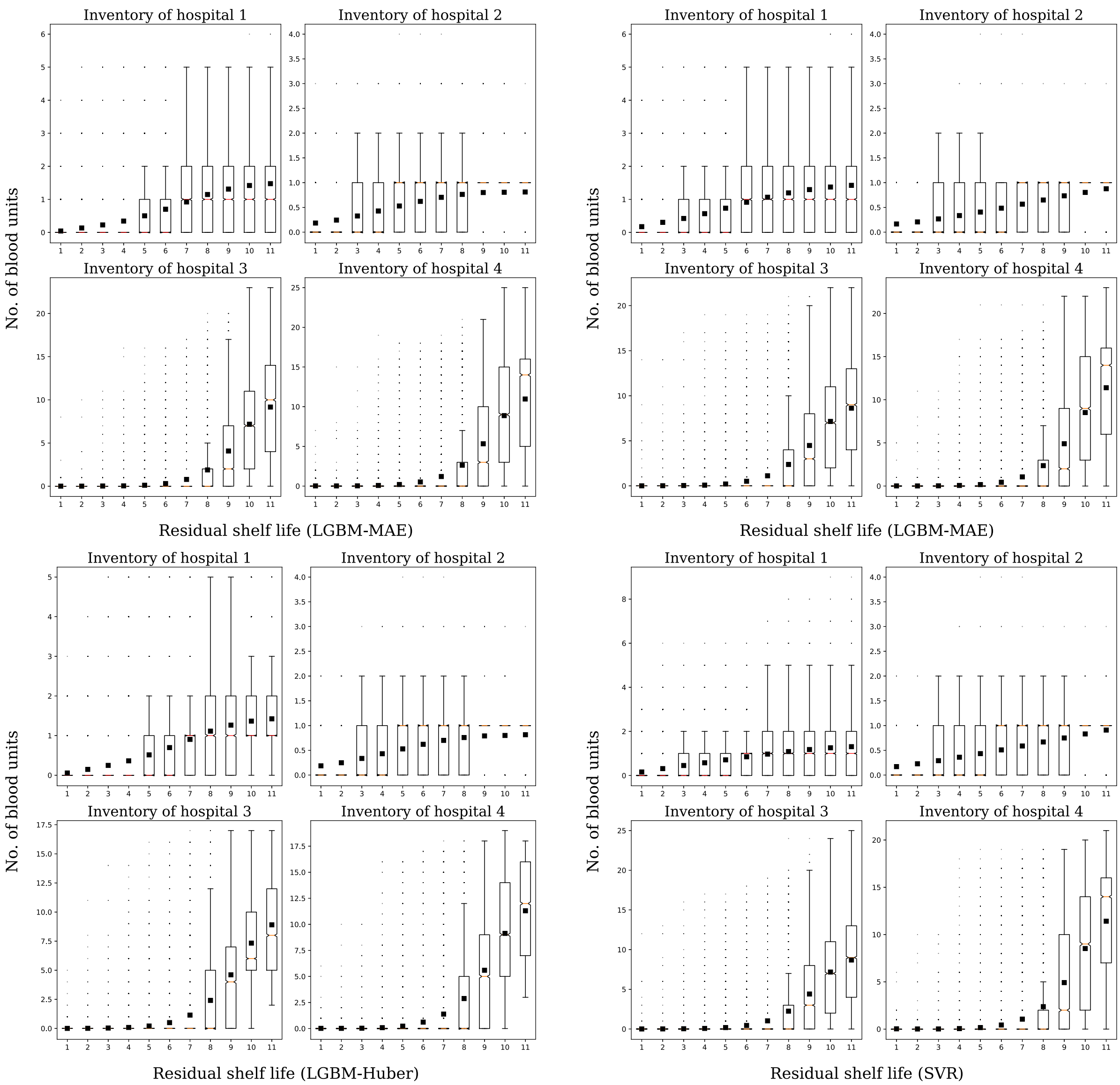

Fig. B.1. The distribution of inventory level of blood units with different ages at each hospital according to LGBM and SVR models over 9250 days. The inventory level is used to predict the decisions of transshipment and is vitally important in determining the performance and cost of the network.

each hospital and each ML model, are depicted in Figs. A.1 and A.2 for 9250 days as well as the TS model current policy.

As we can see from these graphs, the inventory levels for hospitals differ greatly for blood units of different ages. All hospitals have lower inventory levels for younger blood units. Hospitals 3 and 4 have the highest inventory levels for blood units of 9, 10, and 11 days old. As is evident from the distribution of inventory levels, the LGBM-MAE model consistently generates a lower inventory level for all hospitals. The SVR and ridge models generated a similar level of inventory for the four different hospitals, with the ridge model generating slightly lower levels of inventory. We conclude that the LGBM-MAE model is the best-performing model, on average, across the sample data investigated in our case study. According to the results shown in Table 1, we can assert that the MAE loss function is more appropriate for the blood supply chain problem and similar supply chain problems where we have a linear objective function for minimizing costs.

Appendix B

Table A.1 displays the performance of the ML methods considered in this study for 9250 days as well as the TS and model current policies. The results are reported for all involved costs including holding, transshipment,

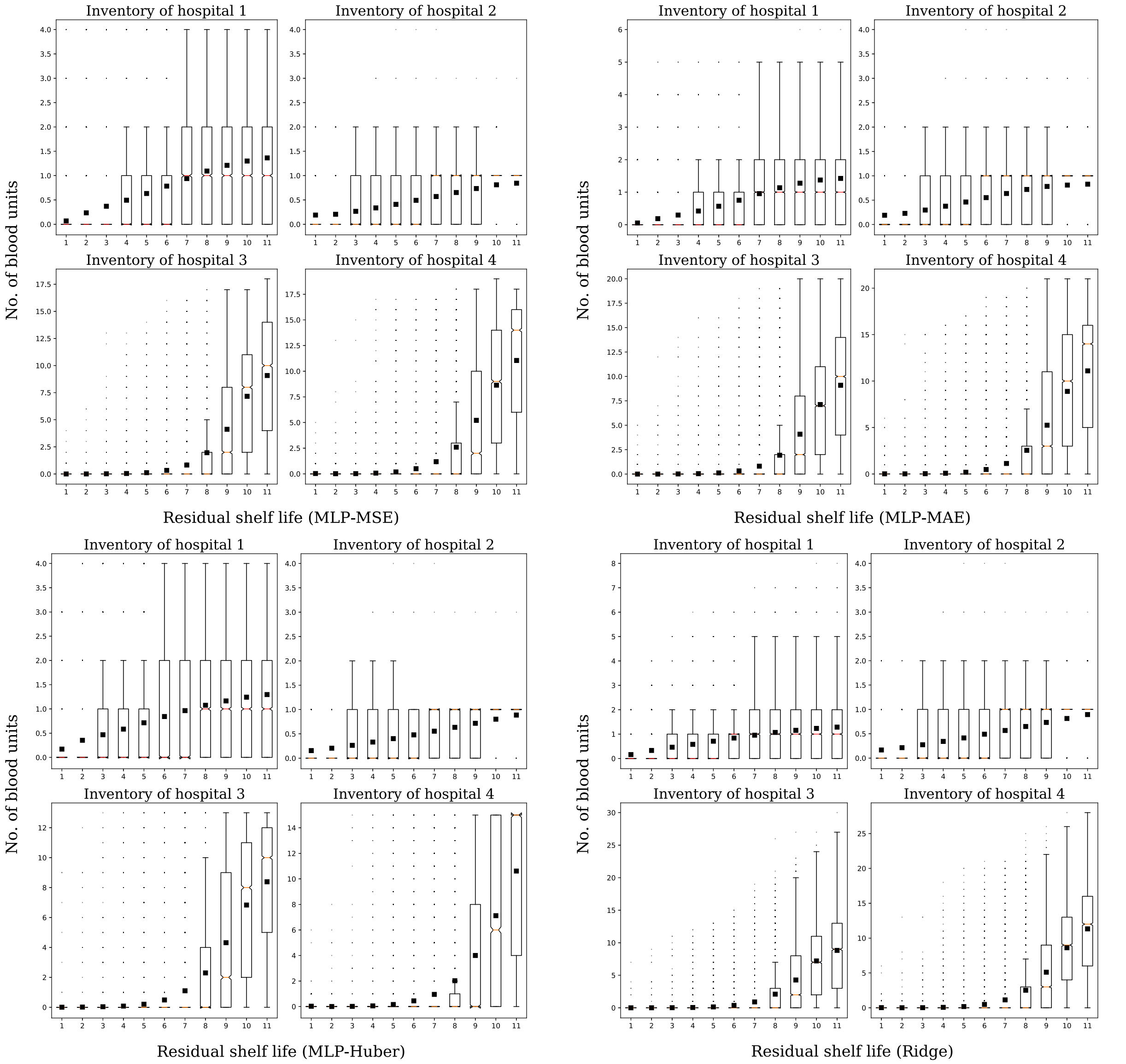

Fig. B.2. The distribution of inventory level of blood units with different ages at each hospital according to MLP and Ridge models over 9250 days. The inventory level is used to predict the decisions of transshipment and is vitally important in determining the performance and cost of the network.

ordering, shortage, and outdate costs. We observe similar results when compared to shorter data with 1850 days. The only notable difference is that the holding cost by LGBM-MSE and MLP-MSE has reduced and the ordering cost has increased. It seems that these models place more orders to the central blood bank and circulate more units, resulting in lower inventory costs and higher ordering costs.

Figs. B.1 and B.2 show the distribution of the inventory level of blood units for each hospital. We observe that inventory levels of blood units are larger for hospitals 3 and 4. Also, we can see that inventory levels are often larger for older blood units. As shown in Fig. B.1 LGBM-MAE model has generated solutions that resulted in a lower level of inventory levels for blood units of different ages, while performing well in terms of the other costs including transhipment, outdate, shortage, and order costs.

References

Abbasi, B., Babaei, T., Hosseinifard, Z., Smith-Miles, K., & Dehghani, M. (2020). Predicting solutions of large-scale optimization problems via machine learning: A case study in blood supply chain management. *Computers & Operations Research*, *119*, Article 104941.

Abbasi, B., & Hosseinifard, S. Z. (2014). On the issuing policies for perishable items such as red blood cells and platelets in blood service. *Decision Sciences*, *45*(5), 995–1020.

Abbasi, B., Vakili, G., & Chesneau, S. (2017). Impacts of reducing the shelf life of red blood cells: A view from down under. *Interfaces*, *47*(4), 336–351.

Abolghasemi, M. (2022). The intersection of machine learning with forecasting and optimisation: theory and applications. arXiv preprint arXiv:2211.13583.

Abolghasemi, M., & Bean, R. (2022). How to predict and optimise with asymmetric error metrics. *arXiv preprint arXiv:2211.13586*.

Abolghasemi, M., Beh, E., Tarr, G., & Gerlach, R. (2020). Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & Industrial Engineering*, *142*, Article 106380.

Abolghasemi, M., & Esmaeilbeigi, R. (2021). State-of-the-art predictive and prescriptive analytics for IEEE CIS 3rd Technical Challenge. arXiv preprint arXiv:2112.03595.

Abolghasemi, M., Hurley, J., Eshragh, A., & Fahimnia, B. (2020). Demand forecasting in the presence of systematic events: Cases in capturing sales promotions. *International Journal of Production Economics*, *230*, Article 107892.

Abolghasemi, M., Tarr, G., & Bergmeir, C. (2022). Machine learning applications in hierarchical time series forecasting: Investigating the impact of promotions. *International Journal of Forecasting*.

Ali, M. M., Boylan, J. E., & Syntetos, A. A. (2012). Forecast errors and inventory performance under forecast information sharing. *International Journal of Forecasting*, *28*(4), 830–841.

Arani, M., Momenitabar, M., Ebrahimi, Z. D., & Liu, X. (2021). A two-stage stochastic programming model for blood supply chain management, considering facility disruption and service level. arXiv preprint arXiv:2111.02808.

Armstrong, J. S. (2001). *Principles of forecasting: A handbook for researchers and practitioners. Vol. 30*. Springer Science & Business Media.

Belien, J., & Force, H. (2012). Supply chain management of blood products: a literature review. *European Journal of Operational Research*, *217*(1), 1–16.

Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, *290*(2), 405–421.

Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, *44*(2), 813–852.

Boylan, J. E., Syntetos, A. A., et al. (2006). Accuracy and accuracy-implication metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, *4*, 39–42.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Chatzos, M., Fioretto, F., Mak, T. W., & Van Hentenryck, P. (2020). High-fidelity machine learning approximations of large-scale optimal power flow. arXiv preprint arXiv:2006.16356.

Clements, M. P., & Hendry, D. F. (1993). On the limitations of comparing mean square forecast errors. *Journal of Forecasting*, *12*(8), 617–637.

Clements, M., & Hendry, D. (1995). On the selection of error measures for comparisons among forecasting methods-reply. *Journal of Forecasting*, *14*(1), 73–75.

Dahl, A. J., Milne, G. R., & Peltier, J. W. (2021). Digital health information seeking in an omni-channel environment: A shared decision-making and service-dominant logic perspective. *Journal of Business Research*, *125*, 840–850.

Dehghani, M., Abbasi, B., & Oliveira, F. (2021). Proactive transshipment in the blood supply chain: A stochastic programming approach. *Omega*, *98*, Article 102112.

Doyle, S. R. (2009). Examples of computing power for zero-inflated and overdispersed count data. *Journal of Modern Applied Statistical Methods*, *8*(2), 3.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, *9*, 155–161.

Elmachtoub, A., Liang, J. C. N., & McNellis, R. (2020). Decision trees for decision-making under the predict-then-optimize framework. In *International conference on machine learning* (pp. 2858–2867). PMLR.

Exterkate, P., Groenen, P. J., Heij, C., & van Dijk, D. (2016). Nonlinear forecasting with many predictors using kernel ridge regression. *International Journal of Forecasting*, *32*(3), 736–753.

Fioretto, F., Hentenryck, P. V., Mak, T. W., Tran, C., Baldo, F., & Lombardi, M. (2020). Lagrangian duality for constrained deep learning. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 118–135). Springer.

Fioretto, F., Mak, T. W., & Van Hentenryck, P. (2020). Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods. In *Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 01* (pp. 630–637).

Fioretto, F., Van Hentenryck, P., W.K. Mak, T., Tran, C., Baldo, F., & Lombardi, M. (2020). A Lagrangian dual framework for deep neural networks with constraints optimization. In *Lecture notes in computer science*: *vol. 12461*, *European conference on machine learning and principles and practice of knowledge discovery in databases* (pp. 118–135). Springer.

Fischetti, M., & Fraccaro, M. (2019). Machine learning meets mathematical optimization to predict the optimal production of offshore wind parks. *Computers & Operations Research*, *106*, 289–297.

Gunpina, S., & Centeno, G. (2015). Stochastic integer programming models for reducing wastages and shortages of blood products at hospitals. *Computers & Operations Research*, *54*, 129–141.

He, H., Daume, H., & Eisner, J. M. (2014). Learning to search in branch and bound algorithms. *Advances in Neural Information Processing Systems*, *27*.

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2022). Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Mining and Knowledge Discovery*.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Hosseinifard, Z., & Abbasi, B. (2018). The inventory centralization impacts on sustainability of the blood supply chain. *Computers & Operations Research*, *89*, 206–212.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*, 3146–3154.

Khalil, E., Dai, H., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. *Advances in Neural Information Processing Systems*, *30*.

Kraus, S., Schiavone, F., Pluzhnikova, A., & Invernizzi, A. C. (2021). Digital transformation in healthcare: Analyzing the current state-of-research. *Journal of Business Research*, *123*, 557–567.

Kruber, M., Lübbecke, M. E., & Parmentier, A. (2017). Learning when to use a decomposition. In *International conference on AI and OR techniques in constraint programming for combinatorial optimization problems* (pp. 202–210). Springer.

Larsen, E., Lachapelle, S., Bengio, Y., Frejinger, E., Lacoste-Julien, S., & Lodi, A. (2022). Predicting tactical solutions to operational planning problems under imperfect information. *INFORMS Journal on Computing*, *34*(1), 227–242.

Levis, A., & Papageorgiou, L. (2005). Customer demand forecasting via support vector regression analysis. *Chemical Engineering Research and Design*, *83*(8), 1009–1018.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Lodi, A., & Zarpellon, G. (2017). On learning and branching: a survey. *TOP*, *25*(2), 207–236.

Makridakis, S., & Spiliotis, E. (2021). The M5 competition and the future of human expertise in forecasting. *Foresight: The International Journal of Applied Forecasting*, (60).

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS One*, *13*(3), Article e0194889.

Mandi, J., Bucarey, V., Tchomba, M. M. K., & Guns, T. (2022). Decision-focused learning: Through the lens of learning to rank. In *International conference on machine learning* (pp. 14935–14947). PMLR.

Meneses, M., Santos, D., & Barbosa-Póvoa, A. (2022). Modelling the blood supply chain–from strategic to tactical decisions. *European Journal of Operational Research*.

Osorio, A. F., Brailsford, S. C., & Smith, H. K. (2015). A structured review of quantitative models in the blood supply chain: a taxonomic framework for decision-making. *International Journal of Production Research*, *53*(24), 7191–7212.

Petropoulos, F., Makridakis, S., Assimakopoulos, V., & Nikolopoulos, K. (2014). 'Horses for Courses' in demand forecasting. *European Journal of Operational Research*, *237*(1), 152–163.

Pirabán, A., Guerrero, W. J., & Labadie, N. (2019). Survey on blood supply chain management: Models and methods. *Computers & Operations Research*, *112*, Article 104756.

Rostami-Tabar, B., Ali, M. M., Hong, T., Hyndman, R. J., Porter, M. D., & Syntetos, A. (2022). Forecasting for social good. *International Journal of Forecasting*, *38*(3), 1245–1257.

Stanger, S., Wilding, R., Hartmann, E., Yates, N., & Cotton, S. (2013). Lateral transshipments: an institutional theory perspective. *International Journal of Physical Distribution and Logistics Management*, *43*(9), 747–767.

Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S., & Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research*, *252*(1), 1–26.

Syntetos, A. A., & Boylan, J. E. (2006). On the stock control performance of intermittent demand estimators. *International Journal of Production Economics*, *103*(1), 36–47.

Theodorou, E., Spiliotis, E., & Assimakopoulos, V. (2023). Optimizing inventory control through a data-driven and model-independent framework. *EURO Journal on Transportation and Logistics*, *12*, 100103.

Torrado, A. S., & Barbosa-Póvoa, A. (2022). Towards an optimized and sustainable blood supply chain network under uncertainty: A literature review. *Cleaner Logistics and Supply Chain*, Article 100028.

Vaclavik, R., Novak, A., Scha, P., & Hanzlek, Z. (2018). Accelerating the branch-and-price algorithm using machine learning. *European Journal of Operational Research*.

Williams, E. P., Harper, P. R., & Gartner, D. (2020). Modeling of the collections process in the blood supply chain: A literature review. *IISE Transactions on Healthcare Systems Engineering*, *10*(3), 200–211.

Zhou, D., Leung, L., & Pierskalla, W. (2011). Inventory management of platelets in hospitals: Optimal inventory policy for perishable products with regular and optional expedited replenishments. *Manufacturing & Service Operations Management*, *13*(4), 420–438.