https://doi.org/10.1016/j.trb.2023.03.009

A smart predict-then-optimize method for targeted and cost-effective maritime transportation

Xuecheng Tian[a], Ran Yan[a1], Yannick Liu[b], Shuaian Wang[a]

[a]*Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Hong Kong*

[b]*Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Hong Kong*

Abstract

In maritime transportation, port state control (PSC) is the last line of defense against substandard ships. During a PSC inspection, PSC officers (PSCOs) identify ship deficiencies that lead to a ship's detention, which can cause severe economic and reputational losses to the ship operator. Therefore, this study innovatively uses PSC inspection data to design ship maintenance plans for ship operators to minimize the overall operational costs. We identify three types of operational costs associated with each deficiency code: inspection cost, repair cost, and risk cost in the ship operators' decision-making process. The risk cost of a deficiency code is strongly related to the detention contribution of the deficiency items under a deficiency code, as indicated by the feature importance of that code in the random forest (RF) model used to predict detention outcome. To design ship maintenance plans, the sequential predict-then-optimize (PO) method generally solves the optimization problem using the input parameters including the predicted probabilities of having deficiency items under each code and the three types of operational costs. However, the loss function in this two-stage framework does not consider the effect of predictions on the downstream decisions. Hence, we use a smart predict-then-optimize (SPO) method using an ensemble of SPO trees (SPOTs). Each SPOT uses an SPO loss function that measures the sub-optimality of the decisions derived from the predicted parameters. By exploiting the structural properties of the optimization problem analyzed in this study, we show that training an SPOT for this problem can be simplified tremendously by using the relative class frequency of true labels within a leaf node to yield a minimum SPO loss. Computational experiments demonstrate that the SPO-based ship maintenance scheme is superior to other schemes and can reduce the total operating expenses of a ship by approximately 1% over the PO-based scheme and by at least 3% over schemes that do not use ML methods. In the long run, SPO-based ship maintenance plans also improve the efficiency of port logistics by reducing the resources needed for formal PSC inspections and alleviating port congestion.

Keywords:

prescriptive analytics; data-driven optimization; port state control (PSC) inspection; ship maintenance planning

[1] Corresponding author. Email addresses: simontxcheng@163.com (X Tian), ran-angela.yan@connect.polyu.hk (R Yan), hnnyly@gmail.com (Y Liu), wangshuaian@gmail.com (S Wang)

1. Introduction

Maritime transportation is the backbone of international trade (Ng, 2015; Sun and Zheng, 2016). Since 1982, port authorities have used port state control (PSC) to inspect foreign visiting ships as the last line of defense against substandard ships. A ship condition found to be non-compliant with the relevant convention(s) during a PSC inspection is deemed a deficiency (IMO, 2017). When port state control officiers (PSCOs) identify critical deficiencies, the port state may detain the ship and may require the ship to rectify these deficiencies before departing. Most studies of PSC have used ML methods and PSC data to design inspection schemes for port authorities, examining how to implement PSC inspections more efficiently, and discussing the effects of PSC inspections (Yan and Wang, 2019). However, to the best of our knowledge, no studies have been conducted to examine how ship operators can benefit from using ML technologies to analyze PSC record data.

For ship operators, ship detention is the most negative outcome of a PSC inspection, as it indicates that the ship is in poor condition and is at risk of incidents and accidents. Furthermore, ship detention delays shipping schedules (Yan et al., 2021b). Maritime transportation delays are costly, with each day's delay consuming between 25% to 50% of a ship's cargo value (Wang et al., 2021). Additionally, by undermining the reputation of the ship's flag state, recognized organization, and company, ship detention leads to higher inspection rates of their ships in the future (Yan et al., 2021b). Hence, given the impact of ship detention on safety, cost, and reputation, being able to identify high-risk ships that have a greater number of deficiencies and conduct targeted ship maintenance before formal PSC inspections is of profound value to ship operators. Furthermore, targeted ship maintenance is not only beneficial to ship operators but can improve overall ship conditions, thereby reducing the resources needed for formal PSC inspections (Xu and Lee, 2018). Reducing the number of detained ships will also alleviate port congestion and improve the efficiency of port operations (Wang et al., 2018). In the long run, targeted ship maintenance will improve maritime safety, the marine environment, and the living and working conditions of the ships' crew, thus achieving the objective of PSC. Therefore, there is a need for studies using PSC data to design ship maintenance schemes that focus on ship operators' decisions about which deficiency codes need to be inspected before formal PSC inspections.

When carrying out ship maintenance before formal PSC inspections, ship operators are concerned with ship deficiencies, which are commonly assumed to be the primary cause of ship detentions (Wu et al., 2021). When ship operators have limited ship maintenance resources, they are likely to prioritize high-risk deficiencies that may warrant detentions. However, to the best of our knowledge, there are no guidelines for detainable deficiencies except for the general descriptions given in International Maritime Organization (IMO) Resolution A.1119(30) (IMO, 2017). Hence, before developing a framework for designing ship maintenance schemes, we first conduct a preliminary analysis to examine the contribution of each deficiency code to the detention outcome—the detention contribution of the deficiency items under each code. We then transform this detention contribution into the risk cost of having deficiency items under each code.

Eruguz et al. (2017) classified the different maintenance strategies as corrective (failure-based), preventive (schedule-based), and predictive (condition-based) maintenance. Our focus on how to design maintenance schemes before formal PSC inspections using ML models and PSC data falls into the third category. Traditionally, the predict-then-optimize (PO) framework is used for the third category, and it consists of two stages: 1) predicting the probability of having deficiency items under each code using ML models and 2) deciding whether to inspect each deficiency code by solving the downstream optimization model built from the predictions. In the PO framework, the prediction model focuses on minimizing the prediction error and ignores the impact of the prediction on the downstream decision. In contrast, this study focuses on the ship operators' objective of obtaining near-optimal ship maintenance decisions while paying less attention to prediction error. We thus train ML models using a loss function that minimizes the decision error by measuring the sub-optimality of the decisions generated by the predictions.

To integrate the prediction task with the optimization task, Elmachtoub and Grigas (2021) proposed a smart predict-then-optimize (SPO) loss function for a broad class of decision-making problems. However, they found that training ML models using SPO loss was likely to be impossible because of the nonconvex and discontinuous characteristics of the SPO loss function. The convex surrogate loss function that they proposed, referred to as SPO+ loss, did not guarantee optimal decisions but only provided an approximation for computational feasibility. In a subsequent study, Elmachtoub et al. (2020) proposed an algorithm for training decision trees using SPO loss, called SPO trees (SPOTs). In our study, we build on this tractable framework proposed by Elmachtoub et al. (2020) and construct tailored SPOTs by exploiting certain structural properties of the optimization problem analyzed in this paper. To improve decision performance, we then train an ensemble of SPOTs referred to as an SPO forest (SPOF).

Our contributions can be summarized as follows. First, our study innovatively uses PSC data to design ship maintenance plans for ship operators. When simulating ship operators' decision-making process, we consider the impacts of ship detention on them and identify three types of operational costs associated with each deficiency code: inspection cost, repair cost, and risk cost. The ship maintenance plans consider all three costs and the occurrence probability of having deficiency items under each code. Second, the risk cost of each deficiency code is determined by the detention contribution of the deficiency items under each code, obtained from a random forest (RF) model of ship detention prediction. Deficiency codes with higher detainable risks are assigned higher risk costs. Third, rather than using the PO framework, we build an SPO framework using an ensemble of SPOTs, which minimizes the SPO loss by leveraging the optimization problem's properties. By comparing our proposed method to other ship maintenance schemes, we demonstrate its superiority.

The remainder of this paper is organized as follows. Section 2 reviews the related research. Section 3 introduces the preliminary concepts, including the ship maintenance planning problem in Section 3.1, the basic introduction about ML models that we use in this paper in Section 3.2, and the method for determining the detention contribution of the deficiency items under each code in Section 3.3. Section 4 depicts the traditional PO framework, and based on this framework, Section 5 proposes the improved SPO framework. Section 6 relates our proposed frameworks to existing studies that predict the optimizer directly rather than the uncertain parameter in the optimization problem. Section 7 describes the computational experiments used to compare different types of ship maintenance schemes. Section 8 concludes the paper and outlines future research directions.

2. Literature Review

Most studies of PSC have focused on improving the inspection efficiency of port authorities, as current ship risk profile (SRP) schemes do not efficiently identify substandard ships. As a PSC inspection can result in the identification of deficiencies and detention, many studies have sought to improve inspection efficiency by developing methods for identifying ships with more deficiencies or higher detention probabilities. For example, to assist the port state in identifying high-risk ships with more deficiencies, Wang et al. (2019) developed a tree augmented naive (TAN) Bayes classifier to predict the number of deficiencies of each ship. Chung et al. (2020) and Yan et al. (2021c) used the Apriori algorithm to determine the type and sequence of deficiency codes that should be inspected. In recent years, ship deficiency prediction models have been used to inform the allocation of scarce inspection resources. That is known as the PSCO scheduling problem and has been studied by Yan et al. (2020) and Yan et al. (2021a). In their research frameworks, Yan et al. (2020) first compared the results of three RF models with different loss functions that predicted ship deficiency numbers under four deficiency categories and then established optimization models for efficiently matching officers' expertise with ship deficiency conditions. Subsequently, Yan et al. (2021a) improved prediction performance by integrating shipping domain knowledge into an XGBoost model, and the downstream PSCO scheduling models were modified to be more consistent with practice.

Xu et al. (2007a) were the first to develop a support vector machine (SVM) model to predict ship detention using both generic and historic factors. A subsequent study by Xu et al. (2007b) enhanced the prediction performance by introducing new features extracted from web mining technology into the SVM model. Gao et al. (2007) further integrated the SVM model with the K-nearest neighbor model and bag-of-words model to predict ship detention. Yang et al. (2018a) proposed a data-driven Bayesian network (BN) model based on TAN learning to predict the ship detention probability of bulk carriers under the Paris Memorandum of Understanding (MoU). To determine the optimal inspection policy for a port, Yang et al. (2018b) incorporated the results of the ship detention prediction model into a game model that considered both port authorities and ship operators. In a recent study, Wu et al. (2022) used an SVM model to predict ship detention, where input features were selected using an analytic hierarchy process (AHP) and a grey relational analysis (GRA) to improve prediction accuracy. Yan et al. (2021b) used a balanced random forest (BRF) model to predict ship detention to address the issues caused by the low-probability detention outcome.

As mentioned above, some recent studies of PSC have integrated prediction and optimization (Yan et al., 2020, 2021a). As we also combine prediction and optimization by training SPOTs, we briefly review the research on training decision trees for personalizing decisions from a finite set of possible options. Kallus (2017) trained trees with a loss function to maximize the effectiveness of the predictions rather than minimize the prediction error. Bertsimas et al. (2019) studied a similar treatment recommendation problem but adopted a weighted loss function to combine prediction and decision error. Elmachtoub et al. (2020) considered a more general class of decision-making problems that could involve a large number of decisions represented by a general feasible region. To train decision trees under SPO loss, they proposed a tractable methodology called SPOTs. They claimed that SPOTs could benefit from the interpretability of decision trees, allowing for an interpretable segmentation of a set of contextual features with different optimal solutions to the optimization problem of interest. In a recent study, Kallus and Mao (2022) also studied how to fit the forest policies in contextual stochastic optimization problems to directly minimize the optimization costs. There are two major differences between Elmachtoub et al. (2020) and Kallus and Mao (2022). The first one is the different splitting rules for the nodes in a tree. Elmachtoub et al. (2020) split the nodes greedily, searching all possible values of all the chosen features to find the optimal split. Kallus and Mao (2022) termed this method the oracle splitting criterion. However, this method may lead to the problem of burdensome re-optimization for every candidate split when applied to large-scale problems. Therefore, Kallus and Mao (2022) proposed a computationally efficient node splitting method by leveraging a second-order perturbation analysis of stochastic optimization for large-scale optimization problems and datasets. It is possible that an increase in training efficiency is at the cost of a decrease in decision quality because Kallus and Mao (2022) used approximate rather than optimal splitting methods. Therefore, we apply oracle splitting rules that do not consume large computational resources for our problem because of its moderate size and structural properties, and we generate high-quality optimal splits. The second difference lies in the methods of prescriptive analytics. In our work, the PO and SPO frameworks still first predict the uncertain parameters in the optimization problem and then derive the final decisions. Hence, our method belongs to the parameter prediction category. However, Kallus and Mao (2022) learned a direct policy mapping from features to the decision without considering the prediction of the uncertain parameters. The method proposed by Kallus and Mao (2022) thus belongs to the optimizer prediction category termed by Bertsimas and Koduri (2021). Moreover, their method is also termed the weighted sample average approximation (w-SAA) method in Bertsimas and Kallus (2020). And in Section 6, we will show that if we consider the same decision loss function when training the ML models no matter using the parameter prediction method (i.e., PO and SPO frameworks) or the optimizer prediction method (i.e., w-SAA method), the decisions prescribed from them are equivalent for our studied problem. Therefore, their prescriptive qualities are the same. However, since the w-SAA method needs to solve a weighted optimization problem considering all training instances when deriving the final decisions, the scale of the weighted optimization problem is much larger than the optimization problem of the PO or SPO framework, leading to a much longer solution time. Therefore, we still adopt the PO and SPO frameworks to reduce solution time.

Our research study contributes to both the literature on PSC and the literature on maintenance and service logistics management. According to Eruguz et al. (2017), our research problem pertains to the subdomains of maintenance strategy selection and maintenance planning. The maintenance strategy selection subdomain focuses on selecting the best maintenance strategy for a system, part, or component to find the optimal balance between the benefits of maintenance and related costs (Eruguz et al., 2017; Goossens and Basten, 2015). The maintenance planning subdomain focuses on maintenance-related tasks, such as inspections, replacements, repairs, and overhauls (Eruguz et al., 2017; Giorgio et al., 2015). As there is no related study using PSC data to design maintenance schemes, our study is a specific application of maintenance and service logistics management research, mainly tailored to PSC inspections. Furthermore, as indicated by the "small amount of failure-related data" (Eruguz et al., 2017, p. 186) characteristic of the maritime sector, scholars face a contradiction where successful preventive maintenance entails preventing the collection of the historical data which we think we need in order to decide what preventive maintenance we ought to be doing (Moubray, 1997). However, with open-source PSC data, the conditions of all inspected ships can be reviewed and shared for academic research. More literature surveys about other subdomains on the topic of maintenance and service logistics in the maritime sector can be also found in Eruguz et al. (2017).

In summary, to the best of our knowledge, there are no existing studies that evaluate the value of PSC data to ship operators, who are directly affected by the lack of attention to operations management in global supply chains (Yang and Qu, 2016). As PSC inspection data are public, ship operators can apply prescriptive analytics methods to these data to improve their decision performance in ship maintenance planning. Identifying detainable deficiencies during ship maintenance would eliminate the adverse impacts of ship detention on ship operators and improve the efficiency of port operations. Therefore, in this study, we address three research gaps. First, the detention contribution of the deficiency items under each deficiency code is innovatively studied to provide useful inputs for the design of ship maintenance plans. Second, a new optimization objective function that minimizes the operational costs of conducting ship maintenance is proposed, which is different from that of previous PSC-related studies. Third, we innovatively apply the SPO framework in providing inspection suggestions for port authorities, which is different from the ML mothods used in previous PSC-related studies.

3. Preliminaries

In this section, we present the ship maintenance planning problem mathematically in Section 3.1. Then, Section 3.2 describes basic information about classification and regression tree (CART) and RF. At last, Section 3.3 introduces how to use the structure of decision trees to measure the detention contribution of the deficiency items under each code, serving as a basis for determining the risk costs associated with the deficiency items under each code.

3.1 Ship maintenance planning problem

We first denote the number of ship deficiency codes as K and an individual ship deficiency code as k $(k=1,...,K)$. A customized ship maintenance plan for an individual ship consists of K recommended decisions, which are denoted as a vector $\mathbf{W}=(w_1,...,w_k,...,w_K)$, where w_k is a binary decision variable that equals 1 if an inspection is recommended for k, and 0 otherwise. To simulate ship operators' decision-making process, we innovatively propose three types of operational costs: inspection cost (c_1), repair cost (c_2), and risk cost (c_3^k).

- The inspection cost c_1 includes manpower and material expenses associated with inspecting items required by a deficiency code. If a ship operator decides to inspect the deficiency items under a deficiency code, it incurs an inspection cost.
- The repair cost c_2 includes the manpower and material expenses if the deficiency items under a code are identified. Considering that identified deficiencies of a ship in a PSC inspection may lead to detention and thus cause huge reputational and economical losses (which will be considered in the risk cost in the following) to a ship operator, we assume that a ship operator conducts repair work if deficiency items are found under each deficiency code. For simplicity, we assume that all deficiency codes have the same inspection cost c_1 and repair cost c_2.
- The risk cost c_3^k represents the loss to a ship operator when a PSCO finds specific deficiency items under code k during a formal PSC inspection. The risk cost of a deficiency code contains two components: indirect and direct costs. Indirect costs refer to the reputational damage to a ship operator from the discovery of deficiency items under code because these identified deficiency items are recorded in the public PSC system. Direct costs refer to the economic costs caused by a delay in the shipping schedule if a ship is detained because of the identified deficiency items. Therefore, we assume that the direct economic costs are related to the detention contribution of the deficiency items under each deficiency code. Recall that there are no specific descriptions of detainable deficiencies except for the general guidelines. In addition, according to the "List of Tokyo MoU Deficiency Codes" (Tokyo MoU, 2019), all the deficiency items considered are extracted from important international regulations and conventions that are proposed to monitor ship conditions from different perspectives. The deficiency items under different codes are guaranteed to be of different nature and are exclusive to each other. As deficiencies under different codes are classified independently, and having any deficiency would definitely increase the detention probability of a ship, their contributions to the detention decision can thus be regarded as independent as well. Therefore, we assume that each deficiency code has a distinct and independent risk cost, derived from the PSC inspection records using the feature importance method, which will be introduced in Section 3.3. Furthermore, although PSCOs can fail to detect all deficiencies because of their negligence and lack of experience or their

personal preference based on domain knowledge, it is difficult to obtain accurate data on this issue because we do not know the ground truth of the inspected ships' deficiency conditions. Actually, such real-life conditions are nearly impossible to obtain because the recording of a deficiency by a PSCO can be highly subjective. We can only use the deficiencies identified and recorded on each ship as the ground truth. Therefore, to fully consider the possible risks that potential deficiencies may bring to ship operators and to reduce the influence of PSCOs' subjectivity, we assume that all items under each code will be inspected and all deficiencies will be identified if a ship is chosen for inspection.

Let us consider the decision for a particular deficiency code k as an example. We let $\mathbf{S}_k = \{0,1\}$ denote the feasible region for the decision of whether to inspect k. The decision-making problem can be defined mathematically as

$$\min_{w_k \in \mathbf{S}_k} z_k(w_k, p_k) = \min_{w_k \in \mathbf{S}_k} \{w_k(c_1 + p_k c_2) + (1 - w_k) p_k (c_2 + c_3^k)\}, \tag{1}$$

where p_k is the probability that a ship has deficiency items under code k. In Equation (1), $c_1 + p_k c_2$ represents the expected costs that a ship operator needs to pay if he/she decides to inspect k, and $p_k(c_2 + c_3^k)$ otherwise. After observing Equation (1), we find that the underlying decision problem always has a unique solution except when $p_k = c_1 / c_3^k$ that leads to two different solutions 0 and 1 with identical objective function values $c_1(1 + c_2 / c_3^k)$. We then let

$$. W_k^*(p_k) = \arg\min_{w_k \in \mathbf{S}_k} \{w_k(c_1 + p_k c_2) + (1 - w_k) p_k (c_2 + c_3^k)\} . \tag{2}$$

denote the set of optimal solutions corresponding to $z_k^*(w_k, p_k)$, and let $w_k^*(p_k)$ denote an arbitrary individual member of the set $W_k^*(p_k)$.

To derive the optimal decision for k, we need to obtain the following parameters: p_k, c_1, c_2, and c_3^k. As mentioned above, costs c_1 and c_2 are assumed to be identical for all deficiency codes and risk cost c_3^k can be determined based on the detention contribution of the deficiency items under each code, elaborated in Section 3.3. Now, only p_k is unknown when solving optimization problem (1). To solve it, the classic approach is a two-stage PO framework, where the first stage uses an ML model that minimizes the prediction error to predict p_k and the second stage solves an optimization problem that integrates p_k with c_1, c_2, and c_3^k. However, this sequential PO framework cannot guarantee that practitioners can obtain near-optimal decisions because the predicting (first) stage focuses on minimizing the prediction error rather than on minimizing the decision error. Our study therefore uses an SPO framework that modifies the loss function used in the ML prediction model to minimize the excess operational costs that result from a (potential) sub-optimal decision under the prediction model over the optimal decision under perfect information. These two frameworks will be introduced in detail in Section 4 and Section 5, respectively.

3.2 Introduction of CART and RF

We mainly use RF and its variant SPOF in this paper. RF is an ensemble of classification and regression trees (CARTs). For one single CART, all of the training examples (data points with input features and output target values) are first stored in the root node. The root node is then split into descendent nodes containing the subsets of all training examples to reduce node impurity, as measured by the Gini index (Breiman, 2001). A split generally depends on a selected feature and one of its values. The splitting rule for classification is to maximize the decrease in the Gini index at each split. Constructing a CART requires splitting the nodes to build a binary decision tree recursively and binarily (Yan et al. 2021b). This process stops when all of the nodes contain training examples of the same output value. However, this may lead to an overfitting decision tree. Therefore, we need to set stopping criteria to prevent trees from becoming too complicated. We use two stopping criteria widely considered to control tree dimension: the maximum depth of a tree (denoted by *max depth*) and the minimum number of training examples per leaf (denoted by *min samples leaf*) (Breiman, 2000). The training process terminates when no node can be further split or one of the stopping criteria is met. For a detailed explanation of CART construction, refer to Breiman (2000) and Yan et al., (2021b).

Ensemble models are used to improve the performance of decision trees. RF is a typical ensemble model that consists of multiple CARTs as weak learners. Compared with a traditional CART, there are two components of randomization in an RF model. The first is that each tree is grown from a bootstrap sample of the original dataset. The second is that a random subset of all the features is chosen to split each node in a tree (Breiman, 2001). Therefore, an RF has two more hyperparameters to ensure the randomness in the RF construction process other than the two hyperparameters used in constructing a single decision tree: the number of CARTs in the RF (denoted by *n estimators*) and the number of features considered in each split (denoted by *max features*) (Breiman, 2000).

3.3 Measuring the detention contribution of the deficiency items under each deficiency code

As mentioned above, we assume that the identified deficiency items under each code have a positive and independent effect on a ship's detention. To assist ship operators in allocating their limited ship maintenance resources to deficiency codes with higher risks, we identify the risk cost of having deficiency items under each code and relate the risk cost to the detention contribution of the deficiency items under each code. Before we introduce the PO and SPO frameworks for solving Optimization Problem (1), we first illustrate how to determine the detention contribution of deficiency items under code k, which provides the criteria for determining the risk cost $k \in \{1,\ldots,K\}$ $\{(\mathbf{x}_i, \mathbf{y}_i, d_i)\}_{i=1}^{n}$ c_3

We adopt the Gini index measure, which is a common feature importance method (Tjoa and Guan, 2021), to capture the detention contribution of the deficiency items under each code, obtained by training an RF model to predict the detention outcome using a given dataset . In this dataset, $\mathbf{y}_i = (y_{1,i},\ldots,y_{k,i},\ldots,y_{K,i})$, where $y_{k,i}$ takes a value of 1 if ship i has deficiency items under code k $(k \in \{1,\ldots,K\})$ and 0 otherwise, and $d_i \in \{1,0\}$ denotes a class label that takes the value of 1 if ship

i is detained and 0 otherwise. Therefore, this preliminary prediction is a classification task whose feature importance can indicate the detention contribution of the deficiency items under each code. In general, any feature in the node that can lead to a large decrease in the Gini index is considered important. Accordingly, the Gini importance of a feature (whether there are deficiency items under code k) in this task is first determined by summing all of the Gini index decreases at the nodes where the split feature .k . falls into the RF and then normalized by the number of input features, namely the total number of deficiency codes.

We now show how to compute the feature importance of code k . The Gini index for node v is defined as:

$$Gini(v) = \sum_{d\in\{0,1\}} rcf(v,d)(1-rcf(v,d)), \tag{3}$$

where $rcf(v,d)$ denotes the relative class frequency for class d in node v and is calculated as follows:

$$rcf(v,d) = \frac{g(v,d)}{\sum_{d'\in\{0,1\}} g(v,d')}, \tag{4}$$

where class $d \in \{0,1\}$ takes the value of 1 if a data example is labelled with a detention outcome and 0 otherwise, and $g(v,d)$ measures the number of training examples belonging to class d that fall into the same node v .

We then let v_L and v_R denote the left and right descendent nodes of node v . The Gini indexes for nodes v_L and v_R are denoted as $Gini(v_L)$ and $Gini(v_R)$, respectively. Then, the Gini index decrease in node v , denoted by $\Gamma(v)$, is calculated as follows:

$$\Gamma(v) = Gini(v) - \left[\frac{N_L}{N}Gini(v_l) + \frac{N_R}{N}Gini(v_R)\right], \tag{5}$$

where N , N_L, and N_R represent the numbers of data examples falling into nodes v , v_L, and v_R , respectively.

To determine the feature importance of k using an RF model, we denote the set $\mathbf{V}_k$ containing nodes in the forest into which split feature k falls. The feature importance of k is denoted by $FI(k)$ and is as follows:

$$FI(k) = \sum_{v\in\mathbf{V}_k} \Gamma(v). \tag{6}$$

Finally, the normalized feature importance of code k , denoted by $NFI(k)$, is as follows:

$$NFI(k) = \frac{FI(k)}{\sum_{k'\in\{1,\ldots,K\}} FI(k')}. \tag{7}$$

4. The PO Framework

Suppose that we have obtained the three types of operational costs c_1, c_2, and c_3^k. To derive $w_k^*(p_k)$ where p_k is unknown, we need to obtain the predicted probability of having deficiency items under code k, namely $\hat{p}_k$. As mentioned above, there are two frameworks for designing ship maintenance plans that combine $\hat{p}_k$ with c_1, c_2, c_3^k. In this section, we focus on designing ship maintenance plans using the PO framework.

In the PO framework, predicting p_k is achieved by training an independent RF model for code k, denoted as f_k, using a given dataset $\{(\mathbf{x}_i, y_{k,i})\}_{i=1}^n$. In this dataset, $\mathbf{x}_i \in \mathbf{R}^q$ denotes a vector of q ship-related features for ship i and $y_{k,i} \in \{1,0\}$ denotes a class label indicating whether ship i has deficiency items under code k. Then, we use the following metric to evaluate the prediction accuracy:

$$l_{Brier}^k := \sum_{i=1}^n (\hat{p}_{k,i} - y_{k,i})^2. \tag{8}$$

This metric is termed Brier score (Brier, 1950) as a strictly proper score function or strictly proper scoring rule that measures the accuracy of probabilistic predictions.

Typically, there are two ways to predict the class probabilities $\hat{p}_k$ using tree ensemble models (i.e., RF), namely the average vote and the relative class frequency (Boström, 2008). The average vote method defines a class probability distribution by averaging the unweighted class votes by the members of the ensemble, where each tree votes for the most probable class. We note that this method does not use the actual class probability distribution of the members, whose predictions may deviate from real probabilities. The relative class frequency method defines a class probability distribution by averaging the relative class frequency of the members of the ensemble. Boström (2008) verified that the relative class frequency method significantly outperforms the average vote method in terms of both accuracy and the area under the receiver operating characteristic (ROC) curve (AUC). However, he also found that the average vote method could give lower Brier score than the relative class frequency. To obtain predicted probabilities based on actual class probability distributions, we select the relative class frequency method to estimate $\hat{p}_k$ of a ship.

Denote the feature vector of a ship to be predicted as $\mathbf{x}$, the trees in the ensemble for predicting $\hat{p}_k$ of the ship as $t_1,...,t_{M_k}$, where M_k denotes the number of trees, and the deficiency condition under code k as y_k. The relative class frequency of y_k in the leaf node where $\mathbf{x}$ falls of an individual decision tree t $(t \in \{t_1,...,t_{M_k}\})$ is denoted as $rcf_k(t,\mathbf{x},y_k)$:

$$rcf_k(t,\mathbf{x},y_k) = \frac{g_k'(t,\mathbf{x},y_k)}{\sum_{y' \in \{0,1\}} g_k'(t,\mathbf{x},y_k')}, \tag{9}$$

where $g_k'(t,\mathbf{x},y_k)$ measures the number of training examples belonging to y_k that fall into the same leaf node as $\mathbf{x}$ in decision tree t. Then, $\hat{p}_k$ can be measured by the relative class frequency RCF_k as follows:

$$\hat{p}_k = RCF_k(\{t_1,\ldots,t_{M_k}\},\mathbf{x},y_k=1) = \frac{\sum_{m=1}^{M_k} rcf_k(t_m,\mathbf{x},y_k=1)}{M_k}. \tag{10}$$

While training the CARTs in the RF for predicting $\hat{p}_k$, we still adopt splitting rules which maximize the Gini index reduction at each split. After the training of f_k is completed, for a new feature vector $\mathbf{x}_{new}$, $\hat{p}_{k,new}$ can be computed by averaging the relative class frequencies of the leaves where $\mathbf{x}_{new}$ falls into. With $\hat{p}_{k,new}$, c_1, c_2, and c_3^k, $w_{k,new}^*(\hat{p}_{k,new})$ can be derived by solving Equation (1).

5. The SPO Framework

As the evaluation metric in the PO framework only measures the prediction error, this section introduces the SPO framework. Section 5.1 demonstrates SPO loss. Section 5.2 introduces a simplified method to train decision trees using SPO loss. Section 5.3 describes the general method for constructing an ensemble of SPOTs.

5.1 SPO loss

By definition, SPO loss is measured not by the prediction error, but by the quality of the decisions that are derived from the predictions. In our problem, the decision error should be measured by the extra operational costs resulting from a (potential) sub-optimal decision under the prediction over the optimal decision made under perfect information. Mathematically, we denote by $\hat{p}'_k$ the predicted probability of having deficiency items under code k obtained from an ML model using SPO loss. We note that the probability distribution of $\hat{p}'_k$ is different from that of $\hat{p}_k$ because of the different splitting rules induced by the adjusted loss function. We then denote by $c_k(w_k^*(\hat{p}'_k),y_k)$ the actual incurred costs that the ship operator needs to pay if the (potential) sub-optimal decision is $w_k^*(\hat{p}'_k)$ when the true label is y_k. For our ship maintenance planning problem, $c_k(w_k^*(\hat{p}'_k),y_k)$ can be in the following four forms:

$$c_k(w_k^*(\hat{p}'_k),y_k) = \begin{cases} c_1, & \text{if } w_k^*(\hat{p}'_k)=1, y_k=0, \\ 0, & \text{if } w_k^*(\hat{p}'_k)=0, y_k=0, \\ c_1+c_2, & \text{if } w_k^*(\hat{p}'_k)=1, y_k=1, \\ c_2+c_3^k, & \text{if } w_k^*(\hat{p}'_k)=0, y_k=1. \end{cases} \tag{11}$$

In the first two forms, if a ship is free from deficiency items under code k ($y_k=0$), and the prescribed optimal decision is to not inspect deficiency items under code k ($w_k^*(\hat{p}'_k)=0$), the ship operator does not incur any cost. However, if $w_k^*(\hat{p}'_k)=1$ and $y_k=0$, a ship operator only has to pay c_1 for an inspection. In the third and fourth forms, if a ship has deficiency items under code k ($y_k=1$), and the prescribed optimal decision is to inspect the deficiency items under code k ($w_k^*(\hat{p}_k')=1$), the ship operator has to pay for the corresponding c_1+c_2. However, if $w_k^*(\hat{p}_k')=0$ and $y_k=1$, the deficiency items under code k may be discovered during a PSC inspection, which leads to a penalty of $c_2+c_3^k$ to the ship operator. In summary, we denote $\mathbf{c}_k=[c_k^1,c_k^2]^T$ as a cost vector where $c_k^1=c_1+c_2y_k$, $c_k^2=(c_2+c_3^k)y_k$ and T represents the transpose of a vector. We then denote $\mathbf{w}_k^*(\hat{p}_k')=[w_k^*(\hat{p}_k'),1-w_k^*(\hat{p}_k')]$ as a prescribed optimal decision vector derived from the prediction $\hat{p}_k'$. Note that $W_k^*(\hat{p}_k')$ may contain more than one optimal solution associated with $\hat{p}'$. Hence, the SPO loss is defined with respect to the worst-case decision from a predicted $\hat{p}'$ as follows:

$$\begin{aligned} l_{SPO}^k(\hat{p}_k',y_k) &= \max_{w_k\in W_k^*(\hat{p}_k')}\left\{c_k(w_k^*(\hat{p}_k'),y_k)\right\}-z_k^*(y_k) \\ &= \max_{w_k\in W_k^*(\hat{p}_k')}\left\{\mathbf{w}_k^*(\hat{p}_k')\mathbf{c}_k\right\}-z_k^*(y_k), \end{aligned} \tag{12}$$

where $\mathbf{w}_k^*(\hat{p}_k')\mathbf{c}_k$ represents the cost incurred from the prescribed optimal decision $w_k^*(\hat{p}_k')$ and $z_k^*(y_k)$ represents the perfect optimal cost if y_k is known (Elmachtoub and Grigas, 2021).

According to Elmachtoub and Grigas (2021), training ML models under the SPO loss Equation (12) is likely to be impossible, as this loss function is nonconvex and discontinuous in terms of the predicted probabilities and the associated operational costs. In this study, inspired by the framework proposed by Elmachtoub et al. (2020), training decision trees under SPO loss $l_{SPO}^k(\hat{p}_k',y_k)$ can be greatly simplified by Theorem 1, which is discussed in the following section.

5.2 Construction of SPOT

We now describe a tractable method to train decision trees under SPO loss based on Theorem 1, which states that letting $\hat{p}_k'$ be equal to the relative class frequency of $y_k=1$ to a leaf node minimizes the SPO loss in the leaf node.

Theorem 1. Let $R_{k,e}$ denote the set of examples falling into the leaf node e for code k, $\bar{y}_{k,e}=rcf_k(t,\mathbf{x},y_k=1)=\dfrac{g_k'(t,\mathbf{x},y_k=1)}{\sum_{y_k'\in\{0,1\}}g_k'(t,\mathbf{x},y_k')}=\dfrac{1}{|R_{k,e}|}\sum_{i'\in R_{k,e}}y_{k,i'}$ denote the relative class frequency of $y_k=1$ within leaf node e, and $\bar{\mathbf{c}}_{k,e}=[c_1+c_2\bar{y}_{k,e},(c_2+c_3^k)\bar{y}_{k,e}]^T$ denote the average cost vector within leaf node e. If Optimization Problem (1) corresponding to $\bar{y}_{k,e}$ has a unique minimizer, then $\bar{y}_{k,e}$ minimizes the within-leaf SPO loss.

Proof: Let $\bar{y}_{k,e}$ be defined as stated in the theorem. We next show that the within-leaf SPO loss with prediction $\bar{y}_{k,e}$ is a lower bound of that of any other prediction $\hat{p}_{k,e}'$.

The following holds for any $\hat{p}_{k,e}'$:

$$
\begin{aligned}
&\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}} l_{SPO}^{k}(\overline{y}_{k,e},y_{k,i'})-\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}} l_{SPO}^{k}(\hat{p}'_{k,e},y_{k,i'})\\
&=\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}} \max_{w_{k,e}\in W_{k,e}^{*}(\overline{y}_{k,e})}\left\{\mathbf{w}_{k,e}\mathbf{c}_{k,i'}\right\}-\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}} \max_{w_{k,e}\in W_{k,e}^{*}(\hat{p}'_{k,e})}\left\{\mathbf{w}_{k,e}\mathbf{c}_{k,i'}\right\}\\
&=\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}} \left\{\mathbf{w}_{k,e}^{*}(\overline{y}_{k,e})\mathbf{c}_{k,i'}\right\}-\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}} \max_{w_{k,e}\in W_{k,e}^{*}(\hat{p}'_{k,e})}\left\{\mathbf{w}_{k,e}\mathbf{c}_{k,i'}\right\}\ \text{(Because } W_{k,e}^{*}(\overline{y}_{k,e})=\{w_{k,e}^{*}(\overline{y}_{k,e})\}\text{ is a singleton)}\\
&\leq \mathbf{w}_{k,e}^{*}(\overline{y}_{k,e})\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}}\mathbf{c}_{k,i'}-\max_{w_{k,e}\in W_{k,e}^{*}(\hat{p}'_{k,e})}\left\{\frac{1}{\left|R_{k,e}\right|}\sum_{i'\in R_{k,e}}\mathbf{w}_{k,e}\mathbf{c}_{k,i'}\right\}\\
&=\mathbf{w}_{k,e}^{*}(\overline{y}_{k,e})\overline{\mathbf{c}}_{k,e}-\max_{w_{k,e}\in W_{k,e}^{*}(\hat{p}'_{k,e})}\left\{\mathbf{w}_{k,e}\overline{\mathbf{c}}_{k,e}\right\}\\
&\leq 0\ \text{(by the definition of } w_{k,e}^{*}(\overline{y}_{k,e})\text{)}\ .
\end{aligned}
$$

We show above that $\overline{y}_{k,e}$ achieves a minimum within-leaf SPO loss, thereby proving the theorem. □

Recall that $\overline{y}_{k,e}$ only has two optimal solutions in its corresponding decision problem when it equals c_1 / c_3^k. Empirically, to guarantee the uniqueness of the optimal solution given $\overline{y}_{k,e}$, we can add a small noise term to every cost vector in the training set (Elmachtoub et al., 2020). Therefore, we assume that $W_{k,e}^{*}(\overline{y}_{k,e})$ is a singleton for any feasible $\overline{\mathbf{c}}_{k,e}$ in what follows.

Under the SPO framework and with the utilization of Theorem 1, the objective of any decision tree training algorithm is to partition the training examples into L_k leaves for code k, and then the training examples in the L_k leaves are represented as $R_{k,1},\ldots,R_{k,l},\ldots,R_{k,L_k} := R_{1:L_k}$, whose predictions minimize the following loss function:

$$
\min_{R_{1:L_k}\in \mathrm{T}_k}\frac{1}{n}\sum_{e=1}^{L_k}\sum_{i\in R_{k,e}}\left(\mathbf{w}_{k,e}^{*}(\overline{y}_{k,e})\mathbf{c}_{k,i}-z_k^{*}(y_{k,i})\right), \tag{13}
$$

where the constraint $R_{1:L_k}\in \mathrm{T}_k$ requires that the allocation of examples to leaves, $1:L_k$, follows the structure of the decision tree, and is determined through repeated splits of the feature components (Elmachtoub et al., 2020). We next use the same procedure as in CARTs to find a reliable and quick solution to Optimization Problem (13); that is, we use the recursive partitioning method to find the decision tree that minimizes the decision error in the training set. Define $x_{i,j}$ as the *j*th input feature component corresponding to the *i*th training example. Beginning with the entire training set, consider a decision tree split (j_k, s_k) that represents a splitting feature component j_k and a splitting point s_k to partition the training examples into a left child node l and a right child node r for code k:

$$
R_{k,l}(j_k,s_k)=\{i\in\{1,2,\ldots,n\}\,|\,x_{i,j}\leq s_k\}\ \text{ and }\ R_{k,r}(j_k,s_k)=\{i\in\{1,2,\ldots,n\}\,|\,x_{i,j}>s_k\},
$$

if feature j_k is numeric, and

$$
R_{k,l}(j_k,s_k)=\{i\in\{1,2,\ldots,n\}\,|\,x_{i,j}=s_k\}\ \text{ and }\ R_{k,r}(j_k,s_k)=\{i\in\{1,2,\ldots,n\}\,|\,x_{i,j}\neq s_k\},
$$

if feature j_k is categorical. The first split of the decision tree is chosen by computing the pair (j_k, s_k) to minimize the following optimization problem:

$$\min_{j_k, s_k} \frac{1}{n}\left(\sum_{i \in R_{k,l}(j_k, s_k)} \left(\mathbf{w}^*_{k,l}(\overline{y}_{k,l})\mathbf{c}_{k,i} - z^*_k(y_{k,i}) \right) + \sum_{i \in R_{k,r}(j_k, s_k)} \left(\mathbf{w}^*_{k,r}(\overline{y}_{k,r})\mathbf{c}_{k,i} - z^*_k(y_{k,i}) \right) \right), \tag{14}$$

where $\overline{y}_{k,l}$ and $\overline{y}_{k,r}$ represent the relative class frequencies of $y_k = 1$ within the left child node l and the right child node r, respectively.

Optimization Problem (14) can be solved by finding the split that has the minimum objective function value among every possible split (j_k, s_k). From Theorem 1, the objective value of a split can be computed as follows: 1) splitting the training examples according to a possible criteria, 2) determining the relative class frequencies of $y_k = 1$ in two child nodes $\overline{y}_{k,l}$ and $\overline{y}_{k,r}$, and the associated $w^*_{k,l}(\overline{y}_{k,l})$ and $w^*_{k,r}(\overline{y}_{k,r})$, 3) deriving the corresponding incurred costs $c_{k,i}(w^*_{k,l}(\overline{y}_{k,l}), y_{k,i}),\ \forall i \in R_l(j_k, s_k)$ and $c_{k,i}(w^*_{k,r}(\overline{y}_{k,r}), y_{k,i}),\ \forall i \in R_r(j_k, s_k)$ for each example in the two child nodes, and 4) summing the SPO losses of each node and dividing the sum by n. After the first split is chosen, the greedy split selection approach is then recursively applied to the resulting nodes. During the training process, two stopping criteria widely considered in an RF are also applied to control the dimension of an SPOT, namely the maximum depth of a tree (*max depth*) and the minimum number of training examples per leaf (*min samples leaf*) (Breiman, 2000). The training process terminates when no node can be further split or one of the stopping criteria is met.

5.3 Construction of SPOF

To improve the performance of SPOTs, we further consider training an ensemble of SPOTs, denoted as SPOF. The procedure for constructing an SPOF is similar to constructing a classic RF and uses the same two components of randomization. The first component is that each SPOT is grown from a bootstrap sample of the training dataset. The second is that a random subset of all of the features is chosen to split each node in an SPOT. After the training procedure is completed, given a new feature vector $\mathbf{x}_{new}$, the relative class frequencies of $y_k = 1$ in the leaves that $\mathbf{x}_{new}$ falls into in the forest are averaged, and the optimal decision on whether to inspect k is determined based on the three types of operational costs. At last, the SPOF is subject to two more hyperparameters similar to an RF, namely the number of SPOTs contained in the SPOF (*n estimators*) and the number of features considered in each split (*max features*) (Breiman, 2000).

6. A Different but Equivalent Prescriptive Analytics Perspective: Optimizer Prediction

From above analysis, for a new feature vector $\mathbf{x}_{new}$, our aim is to first obtain predictions $\hat{p}_{k,new},\ k \in \{1, \ldots, K\}$ (Note that for simplicity, we do not distinguish between $\hat{p}_k$ and $\hat{p}'_k$ in the following), whereas the SPO framework considers the impact of the predictions on the downstream optimization problems while the PO framework does not. Then, we can obtain the optimal prescribed decisions

$$\hat{w}_k(\mathbf{x}_{new}) \in \underset{w_k \in \mathbf{S}_k}{\arg\min}\, z_k\left(w_k, \hat{p}_{k.new}\right),\ k \in \{1,\ldots,K\}. \tag{15}$$

These two frameworks both involve predictions $\hat{p}_{k.new}$ when prescribing the final decisions $\hat{w}_k(\mathbf{x}_{new})$. Alternatively, skipping the intermediate parameter predictions, we can directly learn the forest-based mapping from features to decisions, namely adopting the optimizer prediction method (Bertsimas and Koduri, 2021). We next prove that, similar to Bertsimas and Kallus (2020) and Kallus and Mao (2022), the decisions prescribed from the PO framework or the SPO framework are equivalent to the decisions mapped from the RF-based policy or the SPOF-based policy, respectively.

Proposition 1. When the objective function $z_k(w_k, p_k)$ is linear in p_k, for a new feature vector $\mathbf{x}_{new}$, the PO framework or the SPO framework can prescribe decisions $\hat{w}_k(\mathbf{x}_{new})$ that are equivalent to the forest-based policy mapping $\hat{w}_k^F(\mathbf{x}_{new})$ defined as follows:

$$\hat{w}_k^F(\mathbf{x}_{new}) \in \underset{w_k \in \mathbf{S}_k}{\arg\min} \sum_{i=1}^{n} \beta_{i,k}(\mathbf{x}_{new}) z_k(w_k, y_{k,i}), \tag{16}$$

$$\beta_{i,k}(\mathbf{x}_{new}) := \frac{1}{M_k} \sum_{m=1}^{M_k} \frac{\mathrm{I}\left[t_m(\mathbf{x}_i) = t_m(\mathbf{x}_{new})\right]}{\sum_{i'=1}^{n} \mathrm{I}\left[t_m(\mathbf{x}_{i'}) = t_m(\mathbf{x}_{new})\right]}, \tag{17}$$

where $t(\mathbf{x}_i) = t(\mathbf{x}_{new})$ represents $\mathbf{x}_i$ and $\mathbf{x}_{new}$ fall into the same node of tree t in the forest, and $\mathrm{I}[\square]$ is the indicator function that equals 1 if the condition is true and 0 otherwise.

Proof: Because $z_k(w_k, p_k)$ is linear in p_k, we can obtain $\beta_{i,k}(\mathbf{x}_{new}) z_k(w_k, y_{k,i}) = z_k\left(w_k, \beta_{i,k}(\mathbf{x}_{new}) y_{k,i}\right)$. Therefore, Equation (16) can be transformed into

$$\hat{w}_k^F(\mathbf{x}_{new}) \in \underset{w_k \in \mathbf{S}_k}{\arg\min}\, z_k\left(w_k, \sum_{i=1}^{n} \beta_{i,k}(\mathbf{x}_{new}) y_{k,i}\right). \tag{18}$$

Substituting $\beta_{i,k}(\mathbf{x}_{new})$ by using Equation (17), we can obtain

$$\begin{aligned}
\sum_{i=1}^{n} \beta_{i,k}(\mathbf{x}_{new}) y_{k,i} &= \frac{1}{M_k} \sum_{m=1}^{M_k} \frac{\sum_{i=1}^{n} \mathrm{I}\left[t_m(\mathbf{x}_i) = t_m(\mathbf{x}_{new})\right] y_{k,i}}{\sum_{i'=1}^{n} \mathrm{I}\left[t_m(\mathbf{x}_{i'}) = t_m(\mathbf{x}_{new})\right]} \\
&= \frac{1}{M_k} \sum_{m=1}^{M_k} \frac{g_k'(t_m, \mathbf{x}_{new}, y_k = 1)}{\sum_{y_k' \in \{0,1\}} g_k'(t_m, \mathbf{x}_{new}, y_k')} \\
&= \frac{1}{M_k} \sum_{m=1}^{M_k} rcf(t_m, \mathbf{x}_{new}, y_k = 1) \\
&= RCF_k(\{t_1, \ldots, t_{M_k}\}, \mathbf{x}_{new}, y_k = 1) \\
&= \hat{p}_{k,new}.
\end{aligned} \tag{19}$$

Then, Equation (18) can be transformed into

$$\hat{w}_k^F(\mathbf{x}_{new}) \in \arg\min_{w_k \in \mathbf{S}_k} z_k\left(w_k, \hat{p}_{k,new}\right), \tag{20}$$

which is equivalent to Equation (15). We thus prove that the decisions $\hat{w}_k(\mathbf{x}_{new})$ prescribed from the PO framework or the SPO framework are equivalent to the forest-based policy mapping $\hat{w}_k^F(\mathbf{x}_{new})$.□

By observing Equation (17), we find that $\beta_{i,k}(\mathbf{x}_{new})$ measures the similarity between historical example $\mathbf{x}_i$ and the new example $\mathbf{x}_{new}$ because a larger $\beta_{i,k}(\mathbf{x}_{new})$ represents that examples $\mathbf{x}_i$ and $\mathbf{x}_{new}$ fall into the same nodes in more trees in the constructed forest. Therefore, $\beta_{i,k}(\mathbf{x}_{new})$ is generally termed "weight" in the literature, and Equation (16) can be seen as a weighted sample average approximation (Bertsimas and Kallus, 2020). By comparing Equations (15) and (16), it is obvious that solving Equation (15) is much easier than solving Equation (16) because Equation (16) considers all training examples. Therefore, in our study, we still use Equation (15) to derive decisions.

7. Experimental Results

Section 7.1 describes the dataset that we use and the settings for each ML model. In Section 7.2, we evaluate the detention contribution of the deficiency items under each code. We then show how to determine the three types of operational costs, especially the risk cost. In Section 7.3, we compare the average total operational costs and the single-code costs of five ship maintenance schemes. In Section 7.4, we conduct a sensitivity analysis of risk costs. In Section 7.5, we analyze feature importance of input features under the SPOF. In Section 7.6, we analyze the trade-off between maintenance cost and detention probability of the PO-based inspection scheme and the SPO-based inspection scheme.

7.1 Data description

The dataset for this study contains 3,026 records of PSC initial inspections during January 2015 to December 2019 period at the Hong Kong Port and the corresponding ship-related factors. Hong Kong Port is a member of the Tokyo MoU that governs the Asia-Pacific region. There are 17 deficiency codes required by the Tokyo MoU (2020), as shown in Table 1. We focus on the first 16 deficiency codes that describe specific inspection items and areas. The PSC inspection records are retrieved from the Asia Pacific Computerized Information System[2] (APCIS) provided by Tokyo MoU, and the ship-related factors are obtained from the World Shipping Register database[3].

The preliminary analysis determines the detention contribution of the deficiency items under each code, so we train an RF model with 16 input binary features indicating whether a ship has deficiency items under each code and the output as the detention. The main work is to design ship maintenance plans that cover these 16 deficiency codes, so we train an ML model for each deficiency code, which inputs 14 features that are strongly related to ship condition in the literature (Yan et al., 2020, 2021b):

[2] https://apcis.tmou.org/public/.

[3] https://world-ships.com/.

ship age, gross tonnage (GT), length, depth, beam, type, ship flag performance, ship recognized organization performance, and ship company performance in Tokyo MoU, last PSC inspection time in Tokyo MoU, the number of ship deficiencies in the last inspection in Tokyo MoU, the number of detentions in all historical PSC inspections, the number of flag changes, and whether a ship has a casualty in last 5 years. We follow the data processing method proposed by Yan et al. (2020; 2021b) for these parameters.

Table 1. Description of deficiency codes in the Tokyo MoU

Code	Meaning	Code	Meaning
D1	Certificates and documentation	D10	Safety of navigation
D2	Structural condition	D11	Life saving appliances
D3	Water/Weathertight condition	D12	Dangerous goods
D4	Emergency system	D13	Propulsion and auxiliary machinery
D5	Radio communication	D14	Pollution prevention
D6	Cargo operations including equipment	D15	International Safety Management (ISM)
D7	Fire safety	D18	Labour conditions
D8	Alarms	D99	Other
D9	Working and living conditions		

Before training ML models, we randomly divide the whole dataset into a training set (80%, 2420 records) and a test set (20%, 606 records). We primarily use the RF or SPOF in this study, and we fix *n estimators* in each ML model to be 200. A tuple of three hyperparameters needs to be tuned for these models: *max depth*, *max features*, and *min samples leaf*. We use a grid search with 5-fold cross validation on the training set to tune these hyperparameters in each ML model. The proposed models are constructed using the training set and their performance is validated by using the test set.

7.2 Measuring each deficiency code's risk cost

The preliminary analysis results show that the dataset is highly imbalanced, as it only contains 100 records showing detention (78 detentions in the training set and 22 detentions in the test set). Thus, instead of using accuracy to evaluate the performance of the developed RF model, we adopt *ROC AUC* as our main metric for evaluating the prediction performance. We follow common practice (e.g., Breiman, 2000; Yan et al., 2021b) to determine the search range and the optimal values for the three hyperparameters, which are shown in Table A1 in the Appendix A. The *ROC AUC* score for the predictor on the test set is 0.97, which verifies that it performs 94% better than random guessing. Then, we output the normalized detention contribution of the deficiency items under each code based on the feature importance method, as shown in Figure 1.

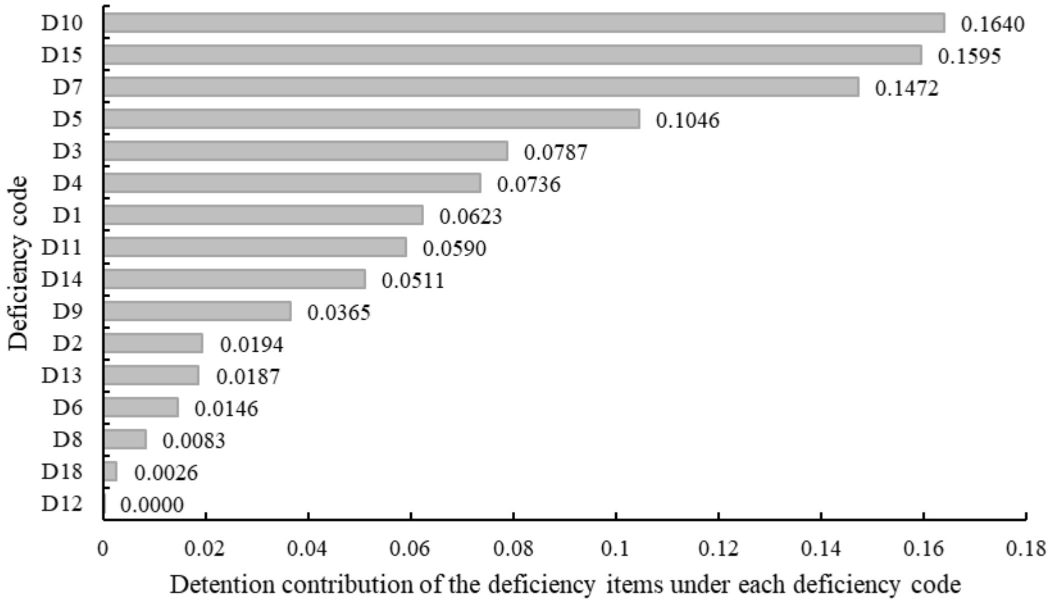

Figure 1. Detention contributions of the deficiency items under 16 deficiency codes

Figure 1 shows that the five deficiency codes with the highest risk are D10, D15, D7, D1, and D3, and the deficiency items under these codes are more likely to lead to detentions. Accordingly, it is reasonable to assign higher risk costs to them. In contrast, the five deficiency codes with the lowest risk are D12, D8, D18, D6, and D13; accordingly, their risk costs are lower. As we cannot obtain accurate values for the three types of operational costs, we measure these costs in units. Following Knapp (2007), who estimated that the average inspection cost is USD 506 when there are no deficiencies and USD 759 when deficiencies (which may involve more procedures and repair work) are found in a PSC inspection at the port, we approximate the inspection and repair costs at two units and three units, respectively (because $506:759 \approx 2:3$). As repair work requires more human and material resources than inspection, it is reasonable to assign a higher value to repair costs than inspection costs. As mentioned in Sections 1 and 3.1, risk cost is divided into reputation cost and detention cost. Reputation cost refers to the reputational damage to a ship operator brought about by the recording of deficiency items in the public PSC information system. The recorded deficiency items indicate that the ship operator cannot guarantee the navigation safety of its ships and can thus lead to higher rates ship inspections in the future because the detention and deficiency history of all ships in an operator's fleet affects the ship operator's performance (Tokyo MoU, 2014). As the effect of detention on reputation is long-lasting and cannot be easily erased once the deficiency items are recorded in the system, we first estimate reputation cost, denoted as c_r, to be five units in benchmark experiments, which is greater than the inspection cost and the repair cost, respectively. We will conduct sensitivity analysis in Section 7.4 to analyze the impact of reputation cost. Next, the detention cost of the deficiency items under each deficiency code is approximated as its detention contribution multiplied by an adjustable factor λ. This adjustable factor converts the normalized detention contribution into detention cost and can be adjusted according to the ship operators' preference. If a ship operator is risk-averse, it would consider improving the risk costs by increasing the value of λ. Conversely, if a ship operator is risk-neutral or risk-appetitive, it would consider decreasing the risk costs by decreasing the value of λ. We will conduct sensitivity analysis in Section 7.4 to analyze the impact of λ. Mathematically, the parameters c_1, c_2, and c_3^k for k are as follows:

$$c_1 = 2 \text{ (units)}, \tag{21}$$

$$c_2 = 3 \text{ (units)}, \tag{22}$$

$$c_3^k = c_r + NFI(k) \times \lambda \text{ (units)}. \tag{23}$$

In our benchmark experiments, we set $c_r = 5$ and $\lambda = 100$, which means that if the deficiency items under a code causes detention, its detention cost is 100 units. Therefore, the detention cost is 50 times greater than the inspection cost, 33.33 times greater than the repair cost, and 20 times greater than the reputation cost. Based on the detention contributions shown above, the risk costs of having deficiency items under each code when $\lambda = 100$ are shown in Figure 2.

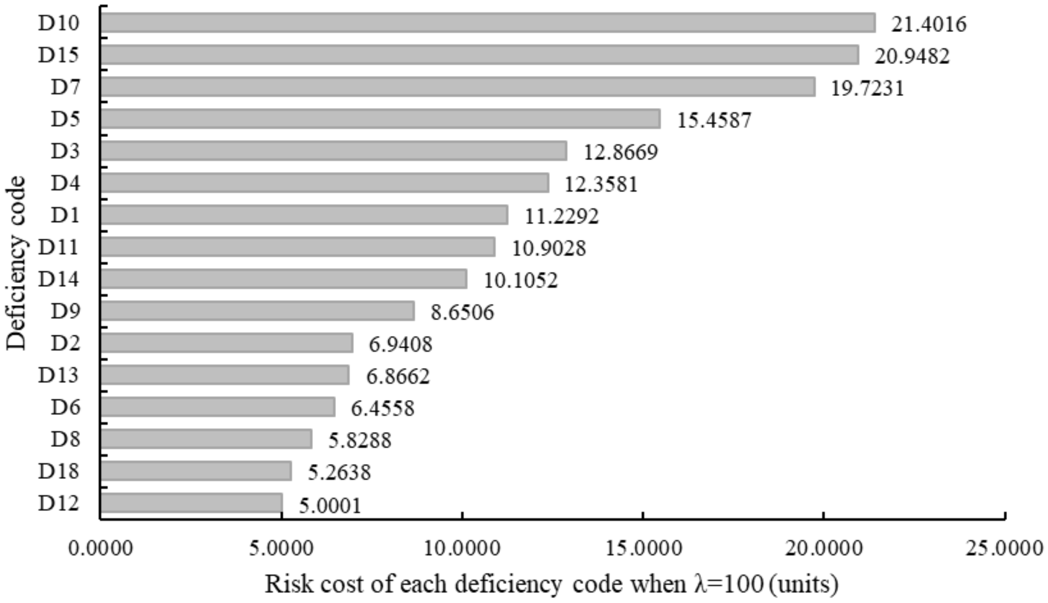

Figure 2. Risk costs of 16 deficiency codes when $\lambda = 100$

7.3 Comparison of ship maintenance schemes

Considering the three types of operational costs, we make ship maintenance decisions for ships in the test set. We compare five different types of ship maintenance schemes. The first scheme is the random inspection scheme (RIS), where each deficiency code has a 50% chance of being inspected. The second scheme is the no inspection scheme (NIS), where ship operators do not perform any ship maintenance. The third scheme is the overall inspection scheme (OIS), where ship operators inspect all deficiency codes. The fourth scheme is the PO scheme (POS). The fifth scheme is the SPO scheme (SPOS). In the benchmark experiments, we set $\lambda = 100$. For RF and SPOF models, we set *n estimators* = 200 and set the search range for *max features* as {2,3,4,5,6,7}, for *max depth* as {2,3,4,5,6,7,8,9,10}, and for *min samples leaf* as {1,2,3,4,5,6,7}. The best hyperparameter tuples for each model are shown in Table A2 in the Appendix A. We then use the optimal hyperparameters to construct each ML model on the whole training set and compute the single-code actual costs $c_k(w_k^*(\hat{p}_k'), y_k)$ incurred by the decision $w_k^*(\hat{p}_k')$ for a ship in the test set. The total incurred operational costs for a ship would be $\sum_{k \in \{1,\ldots,K\}} c_k(w_k^*(\hat{p}_k'), y_k)$. The average total incurred operational costs of five ship maintenance schemes for the test set are shown in Figure 3.

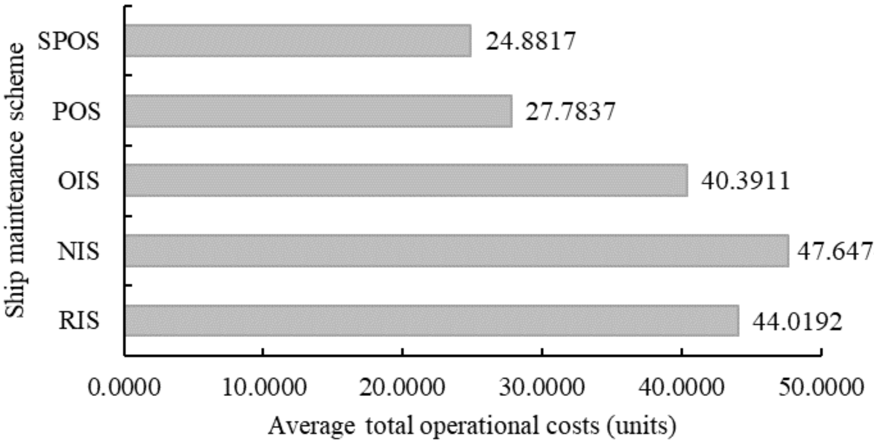

Figure 3. Average total operational costs of five ship maintenance schemes for the test set

As Figure 3 shows, the SPOS outperforms the other four ship maintenance schemes in terms of the average total operational costs. The SPOS reduces the average total operational costs by 43.22% on average compared with ship maintenance schemes that do not apply artificial intelligence methods, such as the RIS, NIS, and OIS. Compared to the POS, the SPOS reduces the average total operational costs by 10.44%, which demonstrates the superiority of SPO loss function that considers decision error. Next, we list the average single-code costs of the five ship maintenance schemes for the test set in Table 2; the proportions indicate the percentage of ships in the test set with deficiency items under each code and gaps 1 to 4 are the cost savings of the SPOS over the RIS, NIS, OIS, and POS, respectively.

Table 2. The average single-code costs of five ship maintenance schemes for the test set

Deficiency	Proportion	Risk cost	Average single-code cost (units)					Gap 1	Gap 2	Gap 3	Gap 4	Standard deviation of 4 gaps
code		(units)	RIS	NIS	OIS	POS	SPOS					
D1	16.72%	11.2292	2.5211	2.5124	2.5297	2.1785	**1.8262**	27.56%	27.31%	27.81%	16.17%	0.06
D2	3.44%	6.9408	1.2349	0.3609	2.1089	0.3609	**0.2520**	79.59%	30.17%	88.05%	30.17%	0.31
D3	23.46%	12.8669	3.3350	3.9275	2.7426	2.6652	**2.5188**	24.47%	35.87%	8.16%	5.49%	0.14
D4	20.22%	12.3581	2.7570	2.9398	2.5743	2.3800	**2.1055**	23.63%	28.38%	18.21%	11.53%	0.07
D5	16.69%	15.4587	2.6466	2.8328	2.4604	2.3385	**1.9129**	27.72%	32.47%	22.25%	18.20%	0.06
D6	1.02%	6.4558	1.0617	0.0936	2.0297	0.0937	**0.0639**	93.98%	31.75%	96.85%	31.80%	0.37
D7	47.32%	19.7231	7.4520	11.3991	3.5050	3.5050	**3.5050**	52.97%	69.25%	0.00%	0.00%	0.36
D8	4.86%	5.8288	1.3123	0.4662	2.1584	0.4663	**0.3078**	76.55%	33.98%	85.74%	33.99%	0.27
D9	28.42%	8.6506	3.2242	3.5375	2.9109	2.8160	**2.5246**	21.70%	28.63%	13.27%	10.35%	0.08
D10	38.60%	21.4016	6.2904	9.4224	3.1584	3.1584	**3.1518**	49.90%	66.55%	0.21%	0.21%	0.34
D11	32.49%	10.9028	3.9984	4.9325	3.0644	3.1248	**2.9447**	26.35%	40.30%	3.90%	5.76%	0.17
D12	0.33%	5.0001	1.0091	0.0132	2.0050	0.0132	**0.0083**	99.18%	37.13%	99.59%	37.12%	0.36
D13	8.96%	6.8662	1.6157	0.9443	2.2871	0.9447	**0.6572**	59.32%	30.40%	71.27%	30.43%	0.21
D14	16.56%	10.1052	2.4484	2.3572	2.5396	2.2037	**1.8305**	25.24%	22.34%	27.92%	16.94%	0.05
D15	6.11%	20.9482	1.8894	1.5807	2.1980	1.2077	**1.0640**	43.69%	32.69%	51.59%	11.90%	0.17
D18	4.49%	5.2638	1.2230	0.3273	2.1188	0.3271	**0.2085**	82.95%	36.29%	90.16%	36.26%	0.29

The findings from Table 2 are as follows. First, the SPOS outperforms the other four ship maintenance schemes on all of the deficiency codes. This result indicates that the average total operational costs of maintaining all of the codes using the SPOS must be lower than the average total costs of randomly adopting the other four schemes to maintain these codes. Second, after analyzing the cost savings of the SPOS over the other four maintenance schemes on all 16 deficiency codes, we find some obvious differences between the deficiency codes related to their proportions and respective risk costs. We further illustrate the values of gaps 1 to 4 of each deficiency code in Figure 4. Note that the purpose of this line chart is not to reflect the development trend of values from gaps 1 to 4, but to highlight the differences and facilitate comparison. The fluctuations in these lines are also used to classify deficiency codes in the following analysis. Moreover, the smoothness of the lines, as indicated by the standard deviation shown in Table 2, is also one of the criteria for classification. Therefore, based on the deficiency codes' proportions, risk costs, gap values, and standard deviations, we roughly classify the deficiency codes into three categories, illustrated using the different lines in Figure 4. Table 3 summarizes the codes in each category and their main characteristics.

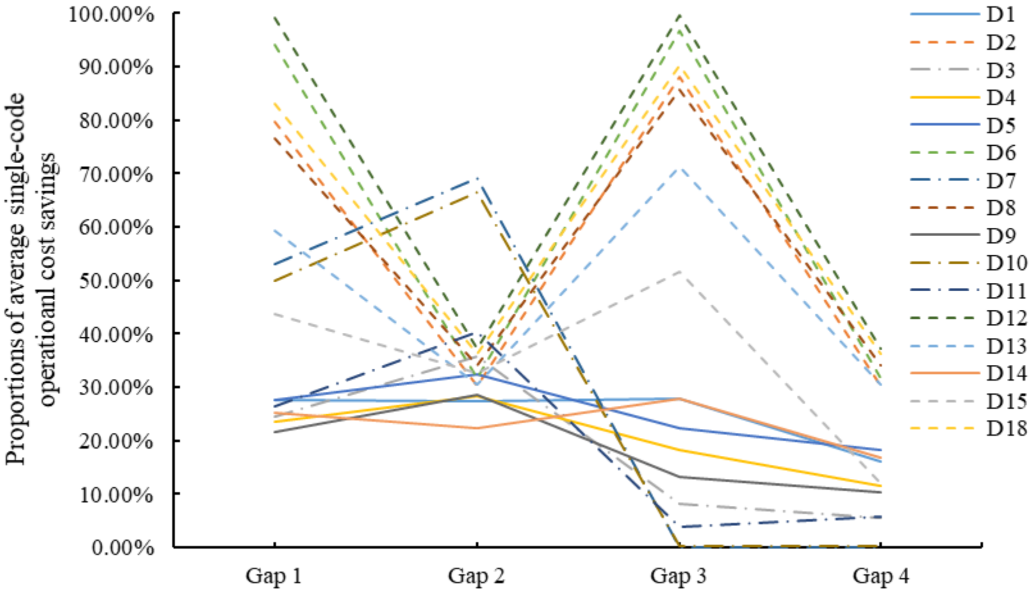

Figure 4. Cost savings of average single-code operational costs for 16 deficiency codes

Category I contains deficiency codes that have a high average proportion and average risk cost and includes D3, D7, D10, and D11. The cost savings from adopting the SPOS are relatively modest for these deficiency codes, especially as indicated by the values of gap 3 and gap 4. This is because cost-conscious and risk-averse ship operators will increase their efforts to inspect these risky deficiency codes and minimize risk costs. Category II contains D1, D4, D5, D9, and D14, which have a medium average proportion and average risk cost. Furthermore, as shown by their smooth lines in Figure 4, the standard deviations of the four gaps of these codes are within 0.1. This result indicates that for these deficiency codes, the SPOS has a relatively stable advantage over the other four ship maintenance schemes. Category III contains D2, D6, D8, D12, D13, D15, and D18, which have a low average proportion and average risk cost. As shown by the values of gap 1 to gap 4, the SPOS can generate significant cost savings for the third category of deficiency codes over the other four schemes. When managing these deficiency codes, ship operators must consider that the NIS may introduce certain risks, while the OIS and the RIS may result in a waste of ship maintenance resources. Although ship operators can use the ML method to predict the probabilities of having deficiency items under these deficiency codes, the POS assigns low predicted probabilities to them because of their low probabilities of occurrence; in this way, the POS prescribes decisions similar to those prescribed by the OIS. Therefore, the superiority of the SPOS is demonstrated by the fact that the minimum cost savings associated with the deficiency codes in category III are greater than 30% compared with the other four schemes.

Table 3. The category of deficiency codes and their characteristics

Category	Deficiency codes	Characteristics
I	D3, D7, D10, D11	High average proportion (35.47%) High average risk cost (16.22 units) “Medium-high-low-low” shape of gap values High average standard deviation (0.25)
II	D1, D4, D5, D9, D14	Medium average proportion (19.72%) Medium average risk cost (11.56 units) Smooth shape of gap values Low average standard deviation (0.06)
III	D2, D6, D8, D12, D13, D15, D18	Low average proportion (4.17%) Low average risk cost (8.19 units) “High-medium-high-medium” shape of gap values High average standard deviation (0.28)

In summary, the SPOS helps ship operators reduce operational costs in ship maintenance for all deficiency codes. Considering the different cost savings brought about by the SPOS for different deficiency codes, we strongly recommend that ship operators invest more monetary and human resources in developing intelligent SPO alarm systems for the deficiency codes in categories II and III because they can always be overlooked due to their medium proportions and risk costs. Installing intelligent alarm systems for these two categories will help ship operators identify these infrequent deficiencies in advance and save on maintenance costs. For the codes in category I, due to their proportions and risk costs, intelligent alarm systems combined with regular maintenance is the best way to prevent detentions.

7.4 Sensitivity analysis of risk costs

We now adjust the values of c_r and λ, respectively, to conduct sensitivity analysis on risk costs. For reputation cost c_r, the candidate value is from set $\{3,5,8,10\}$. For parameter λ, the candidate value is from set $\{50,100,150,200,250\}$. Figures 5 and 6 show the cost savings of the average total operational costs under different values of c_r and λ, respectively. The following observations and conclusions can be drawn. First, the results confirm that the SPOS consistently outperforms the other four schemes. Second, with the increase of c_r and λ, respectively, both gaps 1 and 2 show an upward trend. Intuitively, we see that as the risk costs increase, ship operators pay more for inaction, which is similar to the RIS and NIS. In contrast, both gaps 3 and 4 show a downward trend with the increase of reputation cost and λ, respectively, which means that the cost savings gained from the SPOS over the OIS and POS decrease as risk costs escalate. If the risk costs approach infinity, both the POS and SPOS would resemble the OIS; that is, the best strategy would be to inspect all deficiency codes. Furthermore, as the cost associated with ship maintenance and repair operations is around 10% of the total operating expenses of a ship and it can increase up to 20–30% for old ships (Seaplace, 2020), the SPOS can save approximately 1% more of the total operating expenses than the POS and at least 3% more than the schemes that do not use ML methods. To conclude, the above results verify the superiority of SPOS in reducing the overall operational costs over other schemes.

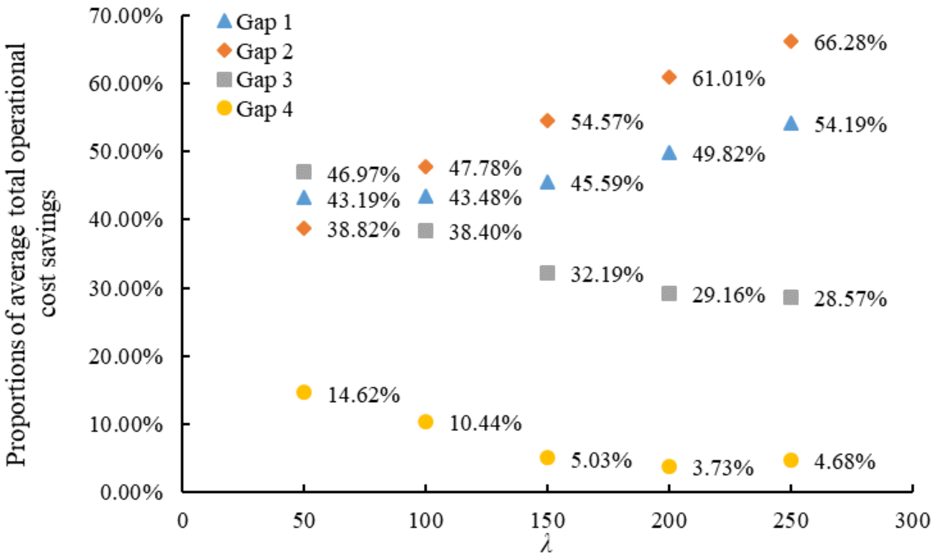

Figure 5. Cost savings of the average total operational costs under different values of λ

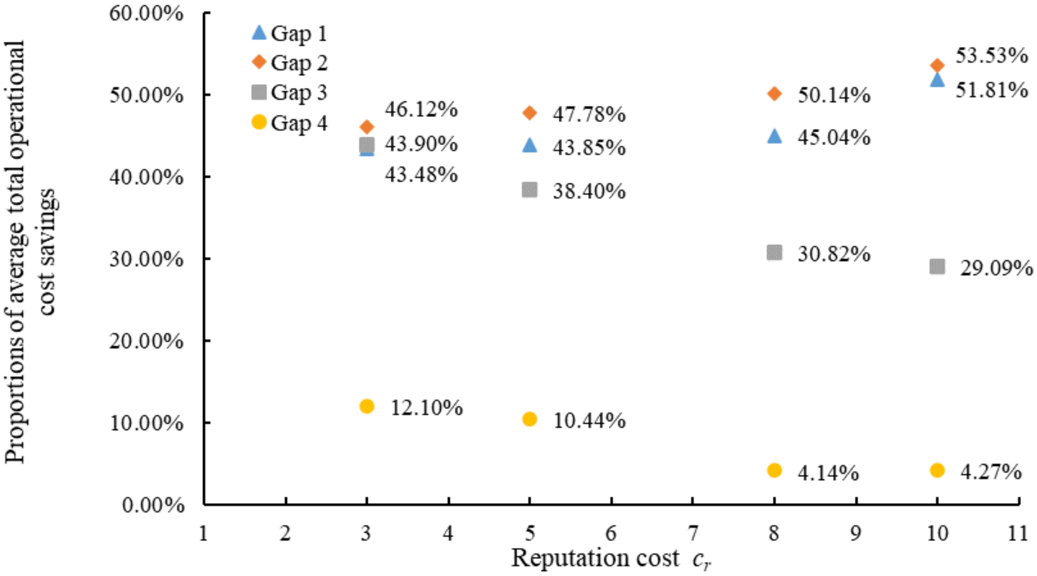

Figure 6. Cost savings of the average total operational costs under different values of reputation cost

7.5 Feature importance of input features under the SPO framework

In this section, we analyze the feature importance of the 14 input features under the SPO framework to ascertain their contribution to ship maintenance decisions. When analyzing the contribution of the 14 input features to ship maintenance decisions, we use SPO loss as the node impurity measure to determine the detention contribution of the deficiency items under each code instead of the Gini index. That is, a feature in the node associated with a greater decrease in the SPO loss is considered more important. Accordingly, the feature importance of an input feature is determined by summing all of the decreases in SPO loss at the nodes where the input feature falls into the RF and is then normalized by the number of input features. Thereafter, we obtain the importance of the input features for each code, as Figure 7 shows. As ship type is a nominal feature and there are six main ship types, we follow Yan et al. (2021a) and use one-hot coding to derive six new binary features when training the ML models. Therefore, the importance of ship type can be weakened, and its real importance value should be greater than that shown in Figure 7.

Regarding these input features, we find that inherent ship features, such as ship age, GT, beam, depth, length, and ship type, play a vital role in the final ship maintenance decisions, which is within our expectations. For example, an older ship would have more deficiency items than a young ship when all of the other conditions are identical or similar. Furthermore, historical PSC inspection records provide valuable information regarding a ship’s performance because features such as last inspection time, last deficiency number, and the number of historical detentions contribute to the final ship maintenance decisions. Last, among the parties considered to be important indicators of ship performance under the new inspection regime of Tokyo MoU, such as flag state performance, recognized organization performance, and ship company performance, only ship company performance shows a significant contribution to the final ship maintenance decisions. This result indicates that there may be a paradox in the criteria set by the Tokyo MoU, which should be investigated further. These findings are valuable to ship operators because they can inform and clarify the features that should be focused on when conducting regular maintenance.

Deficiency code/ Input features	Ship_age	GT	Length	Depth	Beam	Ship_type	NFC*	NDA*	WSC*	SFP*	SROP*	SCP*	LT*	LDN*
D1	0.051	0.140	0.120	0.126	0.158	0.053	0.001	0.045	0.001	0.112	0.079	0.054	0.021	0.041
D2	0.056	0.102	0.113	0.111	0.134	0.089	0.005	0.107	0.001	0.043	0.050	0.055	0.070	0.062
D3	0.139	0.146	0.119	0.154	0.112	0.035	0.002	0.057	0.003	0.014	0.008	0.045	0.090	0.075
D4	0.101	0.186	0.108	0.113	0.103	0.022	0.001	0.037	0.000	0.022	0.001	0.133	0.063	0.112
D5	0.061	0.107	0.115	0.091	0.104	0.152	0.006	0.065	0.006	0.030	0.021	0.089	0.078	0.075
D6	0.092	0.103	0.101	0.093	0.079	0.051	0.004	0.126	0.008	0.071	0.040	0.032	0.093	0.107
D7	0.084	0.151	0.130	0.134	0.135	0.071	0.003	0.050	0.003	0.017	0.010	0.068	0.074	0.070
D8	0.083	0.111	0.125	0.113	0.106	0.074	0.011	0.047	0.010	0.018	0.014	0.041	0.165	0.082
D9	0.061	0.133	0.124	0.133	0.134	0.081	0.005	0.049	0.005	0.026	0.009	0.062	0.086	0.093
D10	0.070	0.135	0.131	0.123	0.124	0.104	0.004	0.042	0.004	0.019	0.008	0.082	0.073	0.083
D11	0.053	0.119	0.132	0.132	0.128	0.140	0.003	0.047	0.004	0.027	0.008	0.074	0.070	0.061
D12	0.081	0.191	0.147	0.091	0.122	0.024	0.003	0.046	0.010	0.000	0.000	0.036	0.158	0.091
D13	0.201	0.043	0.046	0.078	0.052	0.014	0.005	0.186	0.008	0.014	0.001	0.069	0.101	0.184
D14	0.088	0.122	0.136	0.138	0.135	0.064	0.006	0.051	0.005	0.039	0.009	0.069	0.071	0.067
D15	0.059	0.127	0.130	0.100	0.120	0.086	0.002	0.098	0.001	0.040	0.028	0.082	0.056	0.070
D18	0.285	0.099	0.107	0.072	0.121	0.044	0.006	0.012	0.002	0.001	0.004	0.029	0.202	0.015
Average	0.098	0.126	0.118	0.113	0.117	0.069	0.004	0.067	0.004	0.031	0.018	0.064	0.092	0.081

*NFC: the number of flag changes; NDA: the number of detentions in all historical PSC inspections; WSC: whether a ship has a casualty in last 5 years; SFP: ship flag performance; SROP: ship recognized organization performance; SCP: ship company performance; LT: last PSC inspection time in Tokyo MoU; LDN: the number of ship deficiencies in the last inspection in Tokyo MoU.

Figure 7. Feature importance of input features under the SPO framework for different deficiency codes

7.6 The trade-off between maintenance cost and detention probability of POS and SPOS

When characterizing ship operators' inspection decisions, our study assumes that they need to make a trade-off between the maintenance cost (including inspection cost and repair cost) and the risk cost (which is influenced by the detention contribution of having deficiency items under each code). In doing so, the designed optimization problem can be plugged into the training process of SPOF, which shows great superiority in reducing the overall operational costs. To better investigate the real decision process of ship operators, we further divide the designed optimization problem into two stages. First, ship operators make a ship's inspection decision about whether to inspect each deficiency code. If they decide to inspect items under a deficiency code, it would incur an inspection cost. Meanwhile, if inspected items are deficient, we assume that the ship operator would repair them and spend a repair cost. After the inspection and repair work, identified deficiencies would be repaired and the ship's condition would be improved, decreasing its detention probability in the formal PSC inspections. Hence, the inspection decision influences the trade-off between the maintenance cost and the detention probability. It is easy to imagine that an OIS would incur the highest maintenance cost but lead to the lowest detention probability, while a NIS would do it conversely. Following this notion, we assume that a ship operator intends to inspect $k' \in \{1,...,16\}$ deficiency codes randomly using SPOS and POS. Therefore, there would be $2 \times \sum_{k'=1}^{16} \binom{16}{k'} = 131,072$ possible combinations. We then compute the average maintenance cost and the average detention probability of each combination using the test set, where the computing procedure of obtaining the lists of average maintenance cost and detention probability is shown in Appendix B. We then draw trade-off curves to determine the Pareto-optimal combinations based on the obtained lists, which are shown in Figure 8. Under Pareto optimality, a combination is considered dominated if there is another combination that has a lower average maintenance cost or a lower average detention probability (Merdan et al., 2021). We can see that the Pareto frontier formed by the combinations using SPOS almost completely covers the Pareto frontier formed by the combinations using POS. It represents that adopting an SPOS-based inspection combination always leads to a lower maintenance cost and a lower detention probability than adopting a POS-based inspection combination. Therefore, the superiority of SPOF is again verified.

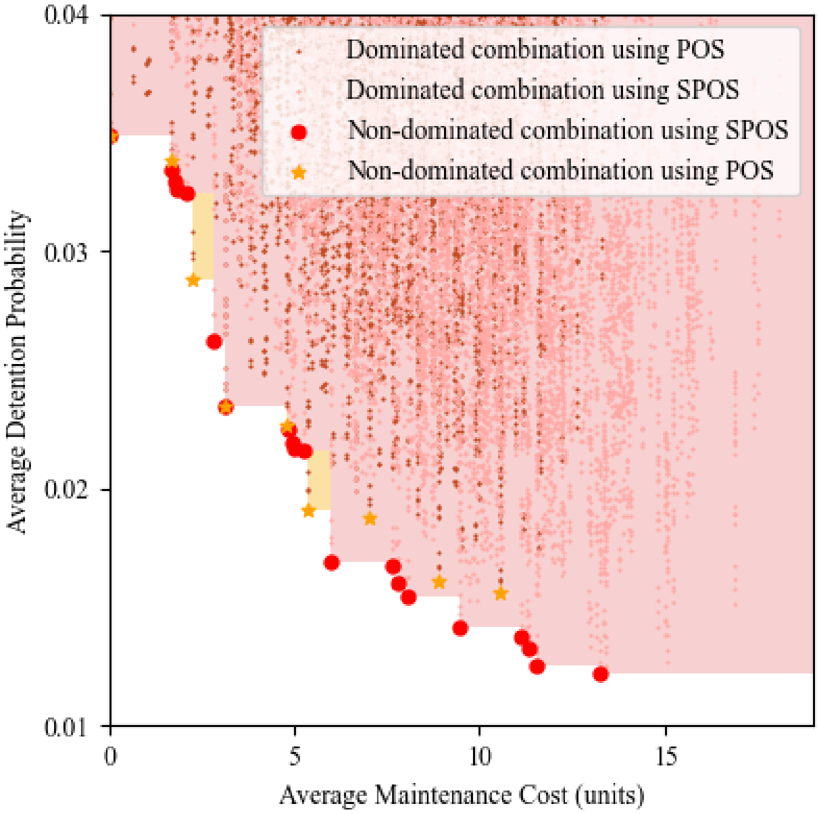

Figure 8. Pareto frontier of inspection combinations using POS and SPOS

8. Conclusions and Future Research Directions

Due to the development of machine learning technologies and the availability of PSC data, many studies have been conducted to improve the efficiency of PSC inspections for port authorities. In contrast, this study designs ship maintenance plans for ship operators with the aim at minimizing the overall operational costs. The targeted and cost-effective ship maintenance plans would also improve the efficiency of port operations by reducing the resources needed for formal PSC inspections and relieving port congestion. When simulating ship operators' decision-making process, we consider the impacts of ship detention on ship operators and innovatively examine three types of operational costs separately: inspection cost, repair cost, and risk cost. In particular, the risk cost of having deficiency items under each code is determined by its detention contribution. Instead of using a PO framework, we propose an SPO framework that adopts an SPO loss function that minimizes the decision error. Computational experiments demonstrate that when the detention cost is 50 times greater than the inspection cost, the average total operational costs derived from the proposed SPO-based scheme are on average 43.22% lower than those derived from ship maintenance schemes that do not use artificial intelligence algorithms and are also 10.44% lower than those derived from the PO-based scheme. Analyses of the average single-code cost savings under the SPO-based scheme indicate that the SPO-based scheme is significantly superior to the other ship maintenance schemes. Furthermore, through a sensitivity analysis of risk costs, we find that the SPO-based scheme can reduce the total operating expenses of a ship by approximately 1 % compared to the PO-based scheme and at least 3% compared to the schemes that do not use ML methods. Finally, the superiority of the SPO framework is further verified by analyzing the trade-off between maintenance cost and detention probability of the SPO-based scheme and the PO-based scheme.

As the first study to propose ship maintenance plans for ship operators, this paper paves the way for future research. First, because of the varied expertise of PSCOs, PSC inspections at different ports have regional characteristics. For example, there may be subjective differences between the PSCOs regarding the criteria for recording a deficiency. In the future, PSC data at multiple ports can be used to generate a more robust ship maintenance plan. Second, future studies could design ship maintenance plans that consider more practical constraints, including but not limited to operational costs, ship maintenance time slot requirements, and the skills of maintenance crews for the deficiency conditions. On the one hand, when formulating the ship maintenance problem, we can further consider the repair decision of ship operators when they make trade-offs between the repair cost and risk cost for an identified deficiency. This consideration may transform the original one-stage optimization problem into a two-stage optimization problem where the solution complexity increases. Under this challenge, model transformation techniques would be needed when the optimization problem is plugged into the training process of ML models under the SPO framework. On the other hand, ship operators can obtain more accurate values of the three types of operational costs considering the requirements of different ship operators and ship types by consulting industrial experts. Third, instead of making inspection decisions separately for each deficiency code, we can further plug the RF model used to predict ship detention probability into the training process of the SPOF. In doing so, we can model the risk cost as a comprehensive value influenced by the overall ship condition to consider the nonlinear relationship between having deficiencies under each code and the final detention outcome. And this comprehensive risk cost would influence the inspection decision backwardly in a coupled manner, and thus further influence the training process of the SPOF. Meanwhile, it is not hard to imagine that this methodology would no doubt need huge computational resources because the studied problem is transformed into a multi-output prescriptive problem with correlated optimization targets, and we need to call the detention probability predictor millions of times for all possible inspection schemes and all possible node splitting rules when training the SPOF. Therefore, we may further consider adopting an approximate splitting rule when constructing the prescriptive trees and reducing the size of inspection scheme pool by heuristics. Third, the SPO framework proposed in this study can be modified for the design of inspection plans for PSCOs in formal PSC inspections by considering the corresponding optimization objectives of PSC authorities. Finally, the SPO framework can be applied to other ML algorithms by taking advantage of their structural features. Their decision-making performance of these alternative ML algorithms for ship maintenance planning should be compared.

Acknowledgment

The authors thank the three reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China [Grant Nos. 72071173, 71831008] and the Research Grants Council of the Hong Kong Special Administrative Region, China [Project number 15201121].

References

Bertsimas, D., Dunn, J., Mundru, N., 2019. Optimal prescriptive trees. INFORMS Journal on Optimization 1(2), 164–183.

Bertsimas, D., Kallus, N., 2020. From predictive to prescriptive analytics. Management Science 66(3), 1025–1044.

Bertsimas, D., Koduri, N., 2021. Data-driven optimization: A reproducing kernel Hilbert space approach. Management Science 70(1), 454–471.

Boström, H., 2008. Estimating class probabilities in random forests. In Proceedings of 2008 International Conference on Machine Learning and Applications, 211–216.

Breiman, L., 2001. Random forests. Machine Learning 45(1), 5–32.

Brier, G., 1950. Verification of forecasts expressed in terms of probability. Monthly Weather Review 78, 1–3.

Calle, M.L., Urrea, V., 2010. Letter to the editor: stability of random forest importance measures. Briefings in Bioinformatics 12(1), 86–89.

Chung, W., Kao, S., Chang, C., Yuan, C., 2020. Association rule learning to improve deficiency inspection in port state control. Maritime Policy & Management 47(3), 332–351.

Elmachtoub, A.N., Grigas, P., 2021. Smart "predict, then optimize". Management Science 68(1), 9–26.

Elmachtoub, A.N., Liang, J.C.N., McNellis, R., 2020. Decision trees for decision-making under the predict-then-optimize framework. In Proceedings of 2020 International Conference on Machine Learning, 2858–2867.

Eruguz, A., Tan, T., Houtum, G., 2017. A survey of maintenance and service logistics management: Classification and research agenda from a maritime sector perspective. Computers and Operations Research 85, 184–205.

Gao, Z., Lu, G., Liu, M., Cui, M., 2008. A novel risk assessment system for port state control inspection. In Proceedings of 2008 IEEE International Conference on Intelligence and Security Informatics, 242–244.

Giorgio, M., Guida, M., Pulcini, G., 2015. A condition-based maintenance policy for deteriorating units. An application to the cylinder liners of marine engine. Applied Stochastic Models in Business and Industry 31(3), 339–348.

Goossens, A., Basten, R., 2015. Exploring maintenance policy selection using the Analytic Hierarchy Process: an application for naval ships. Reliability Engineering and System Safety 142, 31–41.

IMO, 2017. Resolution A.1119(30): Procedure for port state control, 2017. Accessed 28 January 2022.

https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.1119(30).pdf.

Kallus, N., 2017. Recursive partitioning for personalization using observation data. In Proceedings of 2017 International Conference on Machine Learning, 1789–1798.

Kallus, N., Mao, X., 2022. Stochastic optimization forests. Management Science, in press.

Knapp, S., 2007. The econometrics of maritime safety: recommendations to enhance safety at sea. Ph.D. dissertation, Erasmus University Rotterdam.

Merdan, S., Barnett, C., Denton, B., Montie, J., Miller, D., 2021. OR practice–Data analytics for optimal detection of metastatic prostate cancer. Operations Research 69(3): 774–794.

Mokashi, A., Wang, J., Vermar, A., 2002. A study of reliability-centred maintenance in maritime operations. Marine Policy 26, 325–335.

Moubray, J., 1997. Reliability-centered maintenance. Industrial Press Inc. New York.

Ng, M.W., 2015. Container vessel fleet deployment for liner shipping with stochastic dependencies in shipping demand. Transportation Research Part B: Methodological 74, 79–87.

Seaplace, 2020. Ship Maintenance Cost: How can owners reduce it? Accessed 25 February 2022. https://www.seaplace.es/maintenance-cost-how-can-owners-reduce-it/.

Sun, Z., Zheng, J., 2016. Finding potential hub locations for liner shipping. Transportation Research Part B: Methodological 93, 750–761.

Tjoa, E., Guan, C., 2021. A survey on explainable artificial intelligence (XAI) toward medical XAI. IEEE Transactions on Neural Networks and Learning Systems 32(11), 4793–4813.

Tokyo MoU, 2014. Information Sheet of the New Inspection Regime (NIR). Accessed 12 July 2022. http://www. tokyo-mou.org/doc/NIR-information%20sheet-r.pdf.

Tokyo MoU, 2019. List of Tokyo MoU deficiency codes. Accessed 28 March 2022. https://www.tokyo-mou.org/doc/Tokyo%20MOU%20deficiency%20codes%20(December%202019).pdf.

Tokyo MoU, 2020. Annual report on port state control in the Asia-pacific region 2020. Accessed 22 December 2021. http://www.tokyo-mou.org/doc/ANN20-f.pdf.

Wang, S., Yan, R., Qu, X., 2019. Development of a non-parametric classifier: effective identification, algorithm, and applications in port state control for maritime transportation. Transportation Research Part B: Methodological 128, 129–157.

Wang, T., Wang, X., Meng, Q., 2018. Joint berth allocation and quay crane assignment under different carbon taxation policies. Transportation Research Part B: Methodological 117, 18–36.

Wang, T., Meng, Q., Wang, S., Qu, X., 2021. A two-stage stochastic nonlinear integer-programming model for slot allocation of a container shipping service. Transportation Research Part B: Methodological 150, 143–160.

Wu, S., Chen, X., Shi, C., Fu, J., Yan, Y., Wang, S., 2022. Ship detention prediction via feature selection scheme and support vector machine (SVM). Maritime Policy & Management 49(1), 140–153.

Xu, R., Lu, Q., Li, W., Li, K., Zheng, H., 2007a. A risk assessment system for improving port state control inspection. In Proceedings of 2007 International Conference on Machine Learning and Cybernetics, 818–823.

Xu, R., Lu, Q., Li, K., Li, W., 2007b. Web mining for improving risk assessment in port state control inspection. In Proceedings of 2007 International Conference on Natural Language Processing and Knowledge Engineering, 427–434.

Xu, Z., Lee, C.-Y., 2016. New lower bound and exact method for the continuous berth allocation problem. Operations Research 66(3), 778–798.

Yan, R., Wang, S., 2019. Ship inspection in port state control—review of current research. Smart Transportation Systems 2019, 233–241.

Yan, R., Wang, S., Cao, J., Sun, D., 2021a. Shipping domain knowledge informed prediction and optimization in port state control. Transportation Research Part B: Methodological 149, 52–78.

Yan, R., Wang, S., Falgerholt, K., 2020. A semi-"smart predict then optimize" (semi-SPO) method for efficient ship inspection. Transportation Research Part B: Methodological 142, 100–125.

Yan, R., Wang, S., Peng, C., 2021b. An artificial intelligence model considering data imbalance for ship selection in port state control based on detention probabilities. Journal of Computational Science 48, 101257.

Yan, R., Zhuge, D., Wang, S., 2021c. Development of two high-efficient and innovative inspection schemes for PSC inspection. Asia Pacific Journal of Operational Research 38(3), 2040013.

Yang, Z., Qu, Z., 2016. Quantitative maritime security assessment: a 2020 vision. IMA Journal of Management Mathematics 27(4), 453–470.

Yang, Z., Yang, Z., Yin, J., 2018a. Realizing advanced risk-based port state control inspection using data-driven Bayesian networks. Transportation Research Part A: Policy and Practice 110, 38–56.

Yang, Z., Yang, Z., Yin, J., Qu, Z., 2018b. A risk-based game model for rational inspections on port state control. Transportation Research Part E: Logistics and Transportation Review 118, 477–495.

Appendix A. Hyperparameters Tuning

Table A1. Hyperparameters tuning in the RF model for detention probability prediction

Hyperparameters	Search range	Optimal value
max depth	from 2 to 8	8
max features	from 2 to 7	2
min samples leaf	from 2 to 8	3

Table A2. Best hyperparameter tuples for ML models to derive ship maintenance plans

Deficiency code	*max depth*		*max features*		*min samples leaf*	
	RF	SPOF	RF	SPOF	RF	SPOF
D1	2	5	2	2	1	2
D2	2	10	2	6	3	6
D3	8	10	2	4	1	3
D4	4	6	4	6	3	4
D5	7	4	2	2	1	5
D6	2	10	3	6	1	6
D7	9	10	2	6	6	6
D8	3	10	5	6	1	6
D9	8	3	2	5	5	1
D10	10	10	2	5	5	1
D11	5	8	2	3	2	3
D12	10	10	5	6	4	6
D13	2	10	6	6	3	6
D14	2	3	2	3	1	3
D15	2	4	5	2	2	4
D18	9	10	5	6	2	6

Appendix B. Procedure of Obtaing the Lists of Average Maintenance Cost and Detention Probability

```
Input: Predictor for predicting the probability of having deficiency items in code $k$, $k \in \{1,...,K\}$; Predictor for predicting the detention probability; test set $\{(\mathbf{x}_i, \mathbf{y}_i, d_i)\}_{i=1}^{n}$.
Output: The average maintenance list $\text{avg\_cost}$ and the average detention probability list $\text{avg\_det\_prob}$.

Initialize $\text{avg\_cost=[]}$, $\text{avg\_det\_prob=[]}$, $\{\tilde{\mathbf{y}}_i\}_{i=1}^{n} := \{(\tilde{y}_{1,i},...,\tilde{y}_{K,i})\}_{i=1}^{n} = \varnothing$.
For $k' \in \{1,...,16\}$:
    Formulate all combinations containing $k'$ deficiency codes among $K$ deficiency codes, denoted by $Q$.
    For each combination $q \in Q$:
        For deficiency code $k \in q$:
            For test ship $i \in \{1,...,n\}$:
                Prescribe the inspection decision $w_{k,i}^{*}$ for code $k$ of ship $i$ using $\mathbf{x}_i$.
                If $w_{k,i}^{*} = 1$:
                    Incur an inspection cost $c_{1,i} = 2$.
                    If $y_{k,i} = 1$:
                        Repair the deficiencies and let $\tilde{y}_{k,i} = 0$.
                        Incur a repair cost $c_{2,i} = 3$.
                    End
                Else:
                    $\tilde{y}_{k,i} = y_{k,i}$.
                End
            End
        End
        Compute average cost for combination $q$ $\text{avg\_cost}_q = \frac{\sum_{k \in q} \sum_{i=1}^{n} \left(c_{1,i} + c_{2,i} y_{k,i}\right) w_{k,i}^{*}}{n}$.
        Append $\text{avg\_cost}_q$ into $\text{avg\_cost}$.
        For $k \in K \setminus \{q\}$:
            For $i \in \{1,...,n\}$:
                $\tilde{y}_{k,i} = y_{k,i}$.
            End
        End
        For $i \in \{1,...,n\}$:
            Predict the detention probability $\hat{p}_{\text{det},i}$ of ship $i$ $\tilde{\mathbf{y}}_i$.
        End
        Compute the average detention probability for combination $q$ $\text{avg\_det\_prob}_q = \frac{\sum_{i=1}^{n} \hat{p}_{\text{det},i}}{n}$.
        Append $\text{avg\_det\_prob}_q$ into $\text{avg\_det\_prob}$.
    End
End
Output $\text{avg\_cost}$ and $\text{avg\_det\_prob}$.
```

Note: We should train predictors using training dataset before this procedure.