Using forest-based models to personalise ventilation treatment in the ICU

Optimising positive end-expiratory pressure assignment based on the MIMIC-IV dataset

Hubert Nowak[1]

Supervisors: Jesse Krijthe[1]**, Rickard Karlsson**[1]**, Jim Smit**[1,2]

[1]**EEMCS, Delft University of Technology, The Netherlands**
[2]**Department of Intensive Care, Erasmus University Medical Center, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Hubert Nowak
Final project course: CSE3000 Research Project
Thesis committee: Jesse Krijthe, Rickard Karlsson, Jim Smit, Jasmijn Baaijens

Abstract

Positive end-expiratory pressure (PEEP) is one of the components of mechanical ventilation treatment for patients with acute respiratory distress syndrome (ARDS). Correct PEEP level can reduce additional lung injuries sustained during the hospitalisation, significantly increasing patients' chances for survival. In this paper, we focus on estimating the difference in patient mortality when assigned high or low PEEP level. We look at three machine learning models specifically designed for such tasks: S-learner, T-learner and causal forest. Through a series of experiments, we determine their best use cases based on simulated data and measure their performance on a real-life dataset - MIMIC-IV. In our analysis, we find that after tuning the hyperparameters, the models can, to some degree, make valuable predictions and reveal heterogeneity in the treatment effect. However, when evaluated on a separate dataset, the models' performance drops significantly.

1 Introduction

Mechanical ventilation is a crucial point of supportive therapy for critically ill patients who were admitted to the intensive care unit (ICU) with acute respiratory distress syndrome (ARDS) [1]. However, mechanical ventilation can also cause further lung injuries, such as abnormal opening and closing of small airways and alveoli [2]. These issues can be mitigated by using positive end-expiratory pressure (PEEP) [3], which additionally helps increase the area of aerated lung available during inhalation [4], further improving the patient's condition. PEEP can be set to different values, which are categorised into two main groups: high and low. Both of these groups have their advantages and disadvantages, for example, high PEEP levels allow for the usage of a lower fraction of inspired oxygen, reducing adverse pulmonary effects [1]. On the other hand, higher PEEP might increase cardiac pressures and strain on airways, causing new lung injuries, and even possibly leading to cardiac arrest [4]. Therefore, the PEEP setting is crucial in the treatment, since it has a sizeable influence on the patient's condition, and if chosen correctly, can have a considerable potential of increasing their chances for survival.

In the past, there have been several studies aiming to find which PEEP setting is more beneficial for the patients. Some of them (e.g. [5]) suggested that assigning higher PEEP for patients with severe ARDS (those heaving low PaO_2/FiO_2 ratio) reduced their mortality rate. However, more recently, Sahetya and Brower [4] as well as Cavalcanti et al. [2] in their analyses did not arrive at the same conclusions. Rather, they have observed a higher mortality rate among patients with severe ARDS who received higher PEEP setting.

Walkey et al. [6] have conducted a systematic review of outcomes of eight clinical trials comparing strategies using higher PEEP versus lower PEEP levels in patients with ARDS. Oba et al. [7] have carried out a similar analysis of five datasets, three of which were not included in [6]. In both of these studies, the results did not indicate undoubtedly that either of the PEEP setting options has a clear supremacy over the other. Therefore, it is hypothesised that the PEEP strategy should be chosen on a case-to-case basis, with the optimal setting depending on the individual patient's characteristics.

In this paper we investigate whether one can use machine learning techniques to estimate for patients with ARDS their conditional average treatment effect (CATE) - the difference in outcomes in treatment depending on the chosen PEEP regime and patient characteristics. Specifically, we focus on three forest-based meta-learning models - causal forest [8], S-learner [9] and T-learner [9]. The last two are more general models, not strictly limited to being forest-based, so we restrict ourselves to using only random forest as their base learner. We analyse these models and measure their performance using real-world data - MIMIC-IV dataset [10], [11], [12]. Furthermore, based on the results of this analysis, we determine whether they are suitable for making the predictions mentioned above, and verify our claims using data from a randomised controlled trial.

With the rise in popularity of machine learning in recent years, one can find studies aiming to apply various models in the mechanical ventilation setting. For example, in [13] and [14] authors have studied the ability of neural networks and deep learning models trained on MIMIC-III and eICU datasets to correctly predict future oxygenation levels and respiratory system compliance. Based on the same databases, Peine et al. [15] have trained a reinforcement learning model to suggest a suitable mechanical ventilation regime for patients. According to our knowledge, there has not been any research into estimating CATE for PEEP assignment based on the MIMIC-IV data yet.

The remainder of this paper is structured as follows: section 2 introduces related terminology and describes the MIMIC-IV dataset. Section 3 outlines the models and methods used in the experiments. The detailed setup and results of those experiments are presented in section 4. Section 5 provides further analysis of the results. In section 6 we examine the reproducibility and ethical aspects of our study. Lastly, in section 7, the paper is concluded and an overview of possible future work is given.

2 Preliminaries

In this section, we make a short introduction about relevant concepts. First, we give a detailed description of the MIMIC-IV dataset. Next, we describe notions of causal inference and CATE estimation, as well as introduce relevant terminology. After that, three important causal inference assumptions are discussed. Lastly, we motivate our feature selection choices.

2.1 MIMIC-IV dataset

MIMIC-IV is a publicly available database of medical records of the Beth Israel Deaconess Medical Center located in Boston, United States [10]. It contains various information such as patient measurements, diagnoses, applied treatments etc. The data has been deidentified by the authors - all patient identifiers were removed and a random offset has been applied to any dates.

For our purposes, the database has been pre-processed, resulting in a dataset with information about 3941 patients suffering from hypoxemic respiratory failure. There are 24 features provided, including patient characteristics such as vital signs and blood oxygenation levels, as well as laboratory and ventilation settings. Additionally, we are given information about the high vs low PEEP assignment and the treatment outcome (mortality) after 28 days.

2.2 Causal inference and CATE estimation

Causal inference is the process of uncovering cause-effect relationships between different phenomena. That is, through causal inference we try to explain and predict how the value of one variable will change if the value of another variable is altered. This variable whose value is changing is called treatment and in this paper we will denote it as *W*. For example, in our setting, we are trying to predict the hospitalisation outcome of a patient (whether they die or not) and the treatment is expressed as assigning to them a high ($W = 1$) or low ($W = 0$) PEEP level.

In some cases, the treatment might have different effects on different individuals. For example, people from different age groups might have various responses to a given hospitalisation course. In these cases, we say that the treatment is heterogeneous and can use conditional average treatment effect (CATE) as a measure of comparing the treatments. CATE calculates the estimate of the difference of the outcomes when the individual is treated or untreated, conditioned on the characteristics of that individual:

$$\tau(X) = E[Y_1 - Y_0|X]$$

where the meaning of variables and their interpretation in our use case is as follows: X denotes the vector describing the individual, Y_1 and Y_0 are potential outcomes in cases when the individual is respectively treated (high PEEP, $W = 1$) or remains untreated (low PEEP, $W = 0$). Possible values for Y_w are 0 and 1 with $Y_w = 0$ meaning that the patient has died and $Y_w = 1$ denotes that the patient has survived under given treatment $W = w$.

When trying to estimate CATE, one faces several obstacles. The first one (which we describe in regard to our setting, but is also one of the main challenges of causal inference in general) is that for each patient it is possible to observe the outcome under only one of the possible PEEP assignments. This means that it is impossible to know what would be the outcome if the patient was assigned high PEEP instead of low one (or the other way round) as it is highly unlikely to find another person with the exact same characteristics. Because of that, we are always missing half of the information about the potential outcomes.

Another challenge is connected to the existence of confounders. They are variables that influence both the outcome and the treatment assignment [16]. As mentioned in section 1, in past years there have been some papers suggesting that patients with low PaO_2/FiO_2 ratio (called *pf_ratio* in our dataset) benefit from high PEEP (e.g. [5]). Therefore, doctors who were aware of these claims might have based their decision about the PEEP level assignment on this characteristic. Because of that, *pf_ratio* is a confounder – its value influenced PEEP assignment and might also be related to the outcome - the mortality rate. One of the ways of removing confounding is conducting a randomised controlled trial – an experiment in which the treatment is assigned randomly and does not depend on any variables. However, MIMIC-IV is an observational dataset – the researchers had no influence on the treatment and only registered the data. The existence of confounders introduces bias into predictions, as any difference between the treated and untreated groups means that we cannot unquestionably say that the potential discrepancy between outcomes of these groups is exclusively caused by the treatment. Because of that, our predictions might be highly inaccurate unless we properly account for the presence of these confounders.

2.3 Causal inference assumptions

Causal inference requires three main conditions to hold [17]:

- **Consistency** – the way treatment is applied does not influence the outcome, meaning that the result of a given treatment for a given individual is always the same.
- **Conditional exchangeability** – given a set of confounders, if we split our data into groups based on the values of these features, the treatment assignment is independent of the possible outcomes within these groups. Thus one could exchange the treated and untreated parts (again within the groups) without influencing the result of a study, as in that case the probability of a given outcome, under a given treatment, is the same in both parts.
- **Positivity** – for each individual the probability of being assigned to a given treatment level is positive (greater than zero).

It is not guaranteed that these conditions will hold in observational datasets such as MIMIC-IV and they are often difficult to verify. We can try to check whether positivity holds by training a model predicting the probability of assigning a given treatment level to a given sample, that is $P(W = 0|x)$ and $P(W = 1|x)$. We have done so on MIMIC-IV data using k-neighbours classifier and random forest classifier and found that for 3% of the samples both models estimated that one of $P(W = 0|x)$ and $P(W = 1|x)$ is lower than 1%. This result could mean that positivity does not hold in our dataset, or that these samples are outliers and we would need more data points to properly verify this claim.

The remaining two assumptions are trickier to verify. For consistency we need the treatment to be conducted in exactly the same way for each patient and conditional exchangeability essentially means that all possible confounders are accounted for in the data. Therefore it is impossible to precisely verify these claims, and we need to assume that the conditions hold - the treatment was administered each time in the same way and there are no confounders which are not included in our dataset.

2.4 Selected variables

From all available features, we need to select those that either: (1) are confounders, to be able to account for them within the models, or (2) have an impact on the outcome, without influencing or being influenced by the treatment. Based on the literature, talks with the supervisors of this paper and data-driven methods, such as correlation between features and training models predicting treatment and outcome based on feature values, we have chosen to select the following twelve variables:

- As mentioned in subsection 2.2, **pf_ratio** is a confounder. Two variables used to calculate this ratio **po2** and **fio2** are also markers of blood oxygenation levels, and thus could influence both treatment assignment and the outcome, potentially making them confounders.
- **Pco2** is also a blood oxygenation indicator and we mark it as a confounder for the same reason as *po2* and *fio2*.
- **Driving_pressure** and **plateau_pressure** are variables indicating overall patient condition, which could affect doctor's decision about PEEP level.
- **Bilirubin**, **platelets** and **urea** are inflammation markers which can inform about the state of internal organs, influencing treatment course. Moreover, these variables were highly correlated with the outcomes.
- We found that **weight** had a strong influence on predictions of both treatment and outcome, implying that it is a confounder.
- Lastly, out of variables that were not included in the list already, we identified **age** and **minute_volume** to be strong predictors of the outcome.

3 Methodology

We believe that it is a natural assumption that patients with similar characteristics should in real life have similar outcomes for a given treatment. Therefore, it could be expected that many machine learning techniques would be able to identify and model relationships among data of patients with ARDS and make reliable CATE estimations for them. To test this hypothesis, in this paper we focus on forest-based models because of their capacity to handle outliers (patients abnormally responding to a treatment) and their ability to model non-linear relationships. Moreover, such models offer higher accuracy and are more robust to noise when compared to single tree models.

Specifically, we look at three forest-based CATE estimators: S-learner [9], T-learner [9] (with random forest as the base model for these two learners) and causal forest [8] and analyse their performance in several experiments. In this section, we provide a short overview of these models and explain the general setting of performed tests.

3.1 Models

S-learner works by using a single model which takes as its input, apart from the sample features, the treatment assignment without putting any special emphasis on it. We train this model using the whole dataset and obtain an estimator for the function:

$$\mu(x, w) = E(Y|X = x, W = w)$$

Then we can make CATE estimates by computing:

$$\tau_S(x) = \mu(x, 1) - \mu(x, 0)$$

Unlike S-learner, **T-learner** uses two separate models. Each of them is trained on only a part of the data – one on the untreated samples, the other on the treated. Aside from making this data split, the treatment assignment is not used further in the training process. After that, we obtain two corresponding estimators for functions:

$$\mu_0(x) = E(Y|X = x, W = 0)$$

$$\mu_1(x) = E(Y|X = x, W = 1)$$

The CATE estimator is then obtained by combining:

$$\tau_T(x) = \mu_1(x) - \mu_0(x)$$

Causal forest, introduced by Wager and Athey [8], is an extension of random forest developed in [18]. At a high level, a causal forest consists of *B* separate trees, which have to satisfy several constraints, for example each leaf has to contain at least *k* samples from both treated and untreated groups. Wager and Athey in their paper describe two algorithms for creating such trees – double-sample tree and propensity tree. The first one splits training data into two halves, one of which is used to place splits when building the tree, and the other is utilised to make within-leaf estimates. The second algorithm trains a classification tree which predicts treatment assignment *W* while ignoring the outcome *Y*. Once the forest is built, for each tree the treatment effect is estimated by calculating:

$$\mu(x) = \frac{1}{|\{i : W^i = 1, X^i \in L\}|} \sum_{\{i:W^i=1,X^i \in L\}} Y^i - \frac{1}{|\{i : W^i = 0, X^i \in L\}|} \sum_{\{i:W^i=0,X^i \in L\}} Y^i$$

where *L* is the leaf for which $x \in L$ and (X^i, W^i, Y^i) triplets come from the observed data samples in *L*.

After we obtain $\mu_1, \mu_2, \ldots, \mu_B$ estimates from all trees, we can combine them and acquire estimated CATE:

$$\tau_{CF}(x) = \frac{1}{B} \sum_{i=1}^{B} \mu_i(x)$$

It is worth noting that each tree is trained only on a random fraction *s* of all samples *n*, with $s/n \ll 1$. The authors, following [18] and [19], argue that it is better to take an average of predictions from many different trees rather than try to find a single best-performing tree, as it results in a reduction of the variance of predictions.

3.2 Experimental approach

We have conducted several experiments, training the models and measuring their performance in making valuable predictions. Some of those were conducted on simulated data, while others were based on real-life input. Because of that, we had to choose our methods and evaluation criteria accordingly.

Firstly, we have run a simulation study, training and testing models on artificially generated data. By measuring models' performance in specifically designed conditions, we aimed to determine which models excel under which circumstances, in the hope that this would enable us to make accurate predictions about their performance on MIMIC-IV data and other real-life applications. Since in this simulation setup we have control over both response functions, we can accurately calculate the true treatment effect for a given sample. Therefore, as the evaluation criterion, we have decided to use mean squared error (MSE) of the predicted treatment effect, which is also called precision in estimation of heterogeneous effect (PEHE) [20]. Each experiment was repeated several times, each time on a newly generated set of data. We report the averages of outcomes with corresponding confidence intervals.

Unlike in the simulated experiments, in the MIMIC-IV dataset we do not have access to both outcome values, and thus we cannot use MSE to assess predictions. Instead, we chose to use the Qini curve [21], [22], a metric allowing us to evaluate the quality of ranking of samples based on their CATE predicted by the model. The Qini curve plots the difference in the number of responders ($Y = 1$) in the treated and untreated parts of a subset consisting of top *k* samples ordered by their estimated CATE (with the sizes of these parts scaled to be equal) as a function of *k* (the size of this subset). We can then compare this curve to a *random* (in which samples are considered in random order) and *optimal* ones, and calculate the ratio of areas enclosed by curve pairs *random–our* and *random–optimal*, obtaining the area under the curve (AUC) metric. In our setting, the greater this AUC score, the better the model is at properly distinguishing groups of patients – those who benefit from the high PEEP and those who are negatively affected by it.

We have used the Qini curve and the AUC metric when comparing the performance of the three models, as well as in hyper-parameter optimisation. In these experiments, we have split the MIMIC-IV data into training and test sets with a ratio of 70/30. We have not used a separate validation set due to the limited size of the dataset. However, in regard to hyperparameter tuning, the test set plays the role of a validation set – using it we choose optimal parameters for models which then undergo a final evaluation on unseen data from a randomised controlled trial. All tests resulting in a numerical value (AUC score) were repeated numerous times, with a different train/test split in each iteration. We report averages of results, together with their confidence intervals.

As the last evaluation, we performed an experiment using data from a randomised controlled trial (RCT). In this dataset, high and low PEEP was assigned to patients randomly, without any connection to patients' characteristics. Because of that, we can compute the real average treatment effect (ATE) – the difference between averages of outcomes in the treated and untreated groups. We can then compare this real ATE to those estimated by our models (calculated by taking averages of individual CATE estimates). Moreover, as with MIMIC-IV data, we can plot Qini curves and measure the AUC score.

4 Experimental Setup and Results

In this section we will describe in detail the experiments that were conducted on simulated data, as well as MIMIC-IV and RCT datasets. We discuss the setup and tools used for each test and provide their results.

4.1 Simulated data

In total we have conducted seven experiments on simulated data. In each, the samples were generated in the following way:

1. Specify: number of features *d*; propensity score *e(x)* (a function determining the probability that the sample *x* will be treated); the response functions $\mu_0(x)$ and $\mu_1(x)$
2. Simulate the feature vector: $X_i \sim \mathcal{N}(0, I_d)$, where I_d denotes d-dimensional identity matrix
3. Calculate $Y_0 = \mu_0(X_i) + \varepsilon_0$ and $Y_1 = \mu_1(X_i) + \varepsilon_1$, where $\varepsilon_0, \varepsilon_1 \sim \mathcal{N}(0, 1)$ are added to introduce noise into the data
4. Simulate the treatment assignment: $W_i = \text{Bern}(e(X_i))$
5. Set Y_i to Y_0 or Y_1 based on chosen W_i and thus we obtain a sample (X_i, W_i, Y_i)

This way we generated 40000 samples for the training set and 10000 samples for the test set. Each experiment was repeated 50 times.

In this part of our study, following [8], we have used the k-nearest neighbours algorithm as a comparison model. We have made that choice as trees are also nearest neighbours estimators and we wanted to compare our forest-based models with one that also looks at spatial relations between features. Unlike random forests, k-NN is sensitive to feature scaling, however, in our case feature values were already sampled from $\mathcal{N}(0, 1)$, meaning that no additional normalisation was required. The estimates for k-NN were made using the following formula:

$$\tau_{kNN}(x) = \frac{1}{k} \sum_{i \in S_1(x)} Y^i - \frac{1}{k} \sum_{i \in S_0(x)} Y^i$$

where S_1 and S_0 are sets of *k* nearest neighbours of *x* in the treated and untreated groups respectively. In our experiments we have used $k = 10$.

The first six tests were inspired by [9] and [8]. The details of setups off each of them can be found in Table 1. For models, we used default parameters from their implementations[1]. Average mean squared errors of predictions made by models in these experiments are provided in Table 2. The evolution of MSE depending on the size of the training set can be found in Appendix A.

Based on these results we can make several observations about the models. S-learner clearly outperforms other models when there is no treatment effect (simulations 5 and 6). Since it handles treatment like any other feature, the model might not use it to make any splits while building underlying trees, resulting in a correct estimated CATE equal to 0. T-learner performed best in simulation 3, where the response functions were completely independent of each other. The fact that the model uses two separate base learners meant that it was able to estimate these functions with greater precision than other

[1] Random forests for S- and T-learners use Python library *scikit-learn* (https://scikit-learn.org). Causal forest uses *econml* library (https://econml.azurewebsites.net), which implements the model following [23]

Table 1: Details of the setups of the simulation experiments; d - number of features, e(x) - propensity score, $\mu_0(x)$ and $\mu_1(x)$ - response functions

Sim. no.	*d*	*e(X)*	$\mu_0(X)$	$\mu_1(X)$	Remarks
1	10	0.5	$X \cdot \beta$	$\mu_0(X) + 20$	$\beta \sim U([-5;5]^d)$
2	10	0.01	$X \cdot \beta + 5 \cdot \mathbb{I}_{(X_1>0.5)}$	$\mu_0(X) + 8 \cdot \mathbb{I}_{(X_2>0.1)}$	$\beta \sim U([-5;5]^d)$
3	10	0.5	$X \cdot \beta_1$	$X \cdot \beta_2$	$\beta_1, \beta_2 \sim U([1;30]^d)$
4	10	0.5	$\frac{1}{2}\varsigma(X_1)\varsigma(X_2)$	$-\frac{1}{2}\varsigma(X_1)\varsigma(X_2)$	$\varsigma(x) = \frac{2}{1+e^{-12\cdot(x-0.5)}}$
5	10	0.5	$X \cdot \beta$	$\mu_0(X)$	$\beta \sim U([1;30]^d)$
6	10	$\frac{1}{4}(1+\beta_{2,4}(X_1))$	$2 \cdot X_1 - 1$	$\mu_0(X)$	β - beta distribution

Table 2: Outcomes of simulation experiments. For each of the models average MSE is given, together with a 95% confidence interval in parentheses

Sim. no.	S-learner		T-learner		Causal forest		k-NN	
1	4.133	(3.805; 4.461)	4.133	(3.790; 4.476)	2.361	(2.299; 2.423)	5.890	(5.718; 6.062)
2	29.85	(28.97; 30.73)	17.97	(15.26; 20.68)	11.06	(9.09; 13.03)	21.61	(19.03; 24.19)
3	240.8	(212.4; 269.2)	130.4	(122.0; 138.8)	427.5	(406.7; 448.3)	269.5	(258.7; 280.3)
4	2.064	(2.005; 2.124)	2.064	(2.007; 2.121)	2.014	(1.957; 2.071)	2.522	(2.444; 2.600)
5	2.204	(2.141; 2.267)	67.30	(59.05; 75.55)	13.39	(12.27; 14.51)	115.0	(111.7; 118.3)
6	2.008	(1.951; 2.065)	2.107	(2.049; 2.165)	2.020	(1.963; 2.077)	2.504	(2.435; 2.573)

models. Causal forest performed overall well. Its CATE estimates had the lowest or second lowest MSE in almost all simulations (causal forest performed the worst of all models only in simulation 3), showing that it can produce valuable results in various circumstances. All three of our forest-based models have on average outperformed the k-nearest neighbours algorithm, most notably in simulations 4, 5 and 6, and showed that they can uncover the heterogeneity of the causal effects of the treatment.

From these conclusions, one could expect that the T-learner or causal forest performs best on the MIMIC-IV data, as there the response functions are most likely rather unalike, albeit not as independent of each other as it was in the case of simulation 3. To test this hypothesis we have conducted one more simulation experiment, aiming to generate data closely following the real-life samples from MIMIC-IV. In this setting we have $d = 12$ and *e(x)* function specifically designed to introduce confounding in the data (in a similar way as it was done in simulation 6 – by changing propensity score based on feature values), with the expected number of treated samples to be around 12.5% of the whole dataset (matching the treatment ratio of 12.3% in the MIMIC-IV data). To get response functions $\mu_0(x)$ and $\mu_1(x)$ to match the real ones as closely as possible, we use additional models to represent them. For each of those two functions, we have trained, on a corresponding part of the MIMIC-IV data (treated or untreated; with pre-processing steps described in subsection 4.2), three models to predict the patient's mortality Y based on their normalised characteristics X (thus in total obtaining six estimators). These models were: linear regression, k-nearest neighbours regression (with $k = 5$ to capture the relation between closest samples) and decision tree regression (with minimal leaf size equal to 15 to model spatial relations wider than those picked up by k-NN). Then we obtain $\mu_0(x)$ and $\mu_1(x)$ by taking the average of predicted outcomes made by corresponding three models and generate data samples according to the procedure discussed at the beginning of this subsection.

The average mean squared errors in this simulation were as follows: S-learner - 2.047 (95% CI: 1.982; 2.112), T-learner - 2.072 (95% CI: 2.007; 2.137), causal forest - 2.039 (95% CI: 1.975; 2.103), k-NN - 2.220 (95% CI: 2.148; 2.292). Figure 1 depicts the evolution of MSE as we increase the number of samples the models were trained on.

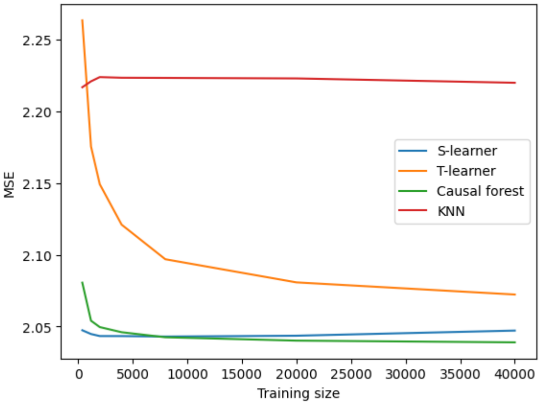


Figure 1: Evolution of MSE in simulation 7

From Figure 1 we can note that both causal forest and T-learner make more accurate predictions the more training data we provide. An interesting observation is that S-learner seems to do the opposite - the MSE of its predictions increases. We have tried to find the cause of this phenomenon and concluded that it is most likely due to the noise we include in our simulations.

4.2 MIMIC-IV

After performing experiments on simulated data, we have conducted multiple tests using the MIMIC-IV dataset.

Firstly we pre-processed the data. We selected features as described in subsection 2.4 and normalised them using z-score normalisation. Missing values were imputed using k-nearest neighbours imputation. Two categorical features were converted into numerical ones – treatment (*peep_regime*): 0 for 'low', 1 for 'high' and outcome (*mort_28*): 1 for 'False' (patient survived), 0 for 'True' (patient died).

The first experiment we ran on the MIMIC-IV data was aimed at making an initial measure of the models' performance on the dataset. We split the data into training and test sets with a ratio of 70/30, train the models with default parameters and calculate the Qini AUC score for both of these sets. We repeated this test 500 times, with different train/test split each time, and measured averages of the results, which can be found in Table 3.

Table 3: Qini AUC scores of models trained on MIMIC-IV with default parameters. In the parentheses we give 95% confidence intervals

Model	Train set		Test set	
S-learner	0.593	(0.54; 0.65)	0.021	(-0.04; 0.08)
T-learner	0.635	(0.61; 0.66)	0.030	(-0.02; 0.08)
Causal forest	0.273	(0.23; 0.32)	0.019	(-0.04; 0.08)

From the results in Table 3, we can notice that the models' relative performance on generated data in simulation 7 does not match their performance on real-life samples. Unlike in the simulation, here T-learner outperformed both S-learner and causal forest, with the last two obtaining comparable scores on the test set. It is worth noting that on the test set all models performed only slightly better than a theoretical model making random CATE estimates (it would have Qini AUC score of 0). Moreover, all models are heavily overfitting, with AUC scores around 15-25 times higher on the training set when compared to the test set.

Trying to mitigate the issue of overfitting and to improve models' performance on the test set we have run a grid search in hopes of finding parameters that would limit the estimators' tendency to closely follow the training set, by for example limiting the depth of trees. We have looked at the following parameters and their possible values:

- **max_depth** - Maximal depth - [3, 5, 20, 40, None], where *None* meant that the trees were expanded as far as possible
- **min_samples_split** - Minimum number of samples required to split an internal node - [5, 10, 20, 35, 50]
- **n_estimators** - Number of trees in the forest - [50, 100, 250, 500] for S- and T-learners; [24, 60, 100, 500, 1000, 2500] for causal forest
- **max_samples** - Fraction of samples used to train each tree - [0.5, 1.0] for S- and T-learners; for causal forest this parameter was always set to the default value of 0.45

We have run this grid search 50 times and in Table 4 we report average Qini AUC scores of the models' versions performing best on the test set. The corresponding best hyperparameters can be found in Appendix B.

Table 4: Qini AUC scores of best performing versions of models found using grid search. In the parentheses we give 95% confidence intervals

Model	Train set		Test set	
S-learner	0.184	(0.13, 0.24)	0.032	(-0.04; 0.10)
T-learner	0.263	(0.22; 0.30)	0.053	(-0.02; 0.12)
Causal forest	0.207	(0.16; 0.25)	0.036	(-0.04; 0.11)

Based on the results in Table 4 we can see that indeed performance of all models has improved, with T-learner still outperforming the other two. Qini AUC score considerably dropped on the training set and increased by around 50%-75% when calculated on unseen data. However, the aforementioned issues that we aimed to solve are still present. The models are still strongly overfitting, and the AUC scores, although greater than before, are still on average not that far from 0. In the last attempt to fix them, we have identified using Shapley values (see subsection 5.2) four features contributing the most to the estimates - *age*, *platelets*, *urea* and *pco2*. We then ran the same grid search again for 20 repetitions, using in our models only these four variables. The results can be found in Table 5 and the hyperparameter values are in Appendix B.

Table 5: Qini AUC scores of best performing versions of models when only *age*, *platelets*, *urea* and *pco2* features were used. In the parentheses we give 95% confidence intervals

Model	Train set		Test set	
S-learner	0.213	(0.17; 0.25)	0.044	(-0.01; 0.10)
T-learner	0.239	(0.21; 0.27)	0.058	(0.01; 0.11)
Causal forest	0.083	(0.04; 0.12)	0.034	(-0.02; 0.09)

The results in Table 5 suggest only a small improvement in Qini AUC scores for S- and T-learners, while causal forest performed worse. Therefore, we can conclude that limiting ourselves to only a subset of features does not significantly increase the performance of the models.

4.3 RCT dataset

The last set of evaluation tests was conducted on the RCT dataset. The data contained 2299 samples, but did not have all features available in the MIMIC-IV database. Particularly, from the list we developed in subsection 2.4 *bilirubin*, *platelets* and *urea* were missing. Because of that, we had to re-train our models, ignoring these three variables. We have done so, using the optimal hyperparameters, on 500 different random train/test splits and chose the ones with the highest Qini AUC score. They were then used to make CATE estimates for available samples.

The real ATE in the dataset was equal to 0.02556. The ATE estimates made by our models were following – S-learner: -0.03194; T-learner: -0.18256; causal forest: -0.11225. The Qini AUC scores for the CATE predictions were as follows – S-learner: 0.00515; T-learner: 0.026117; causal forest: 0.00682. The corresponding Qini curves can be found in Appendix C.

5 Discussion

In this section we further study the outcomes of performed evaluations. Additionally, we mention the limitations encountered in our research.

5.1 Further analysis of the MIMIC-IV tests

Firstly, it needs to be noted that in all of our experiments on the MIMIC-IV data the confidence intervals for our results were quite broad. Because of that, many of our results might be highly inaccurate, for example the found optimal hyperparameters might be wrong or T-learner might not outperform the other two models by such a considerable margin. Similarly, the improvements we have found in models' performance between the experiments might be imprecise.

Secondly, the issue of overfitting persisted throughout the experiments we have performed. Even after tuning hyperparameters and reducing the complexity of models, the Qini AUC score was still 4 to 5 times greater on the train set when compared to the test set.

However, despite these issues, when we plot Qini curves for the models' predictions for the test set, they lay (in most cases) above the random line, indicating that they are, to some extent, capable of distinguishing between groups of patients that benefit and suffer from high PEEP. Figure 2 depicts this situation for the best train/test split we have found among 10 random splits.

One might try to resolve these concerns by running the tests for a greater number of repetitions and with a finer grid search that also includes other hyperparameters. This was unfortunately impossible in our study due to time and computational power constraints and we believe that it still would not completely fix these issues given the rather small absolute value of improvements we were able to achieve in our experiments.

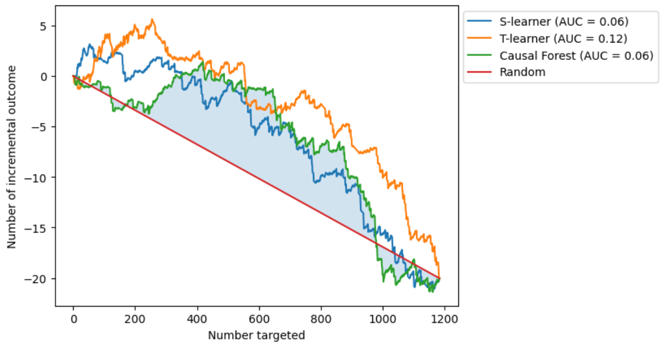


Figure 2: Qini curves for the models with optimal hyperparameters

5.2 Shapley values

Shapley values are a way of explaining black-box models [24]. They provide us with a measure to check which features influence the predictions the most. We have calculated[2] Shapley values for all of our models, both with and without tuned hyperparameters. In almost all of them, *age*, *platelets*, *urea* and *pco2* (not necessarily in this order) emerged as the top 4 most predictive features (only for S-learner with default parameters *pco2* was ranked lower). The detailed plots can be found in Appendix D. According to our knowledge, these four features have not been mentioned before in the literature as variables that help make correct PEEP level assignments, meaning that clinical trials focusing on these features could be conducted, confirming or disproving this correlation.

5.3 Evaluation on RCT dataset

The real ATE on RCT dataset was positive, meaning that on average patients benefited from high PEEP. However, all of our models had negative ATE estimates, predicting that low PEEP was more beneficial. The distance between the real value and the estimates was also significant – up to around 0.2 for T-learner. Moreover, Qini AUC scores were all very close to 0, meaning that the models performed only marginally better than a random baseline.

An unanticipated observation we made is that causal forest predicted a negative treatment effect for all samples, a result that we would not expect in a dataset with positive ATE. This, however, could be a peculiarity of the specific model we used for the evaluation, as it also made negative CATE estimates for all MIMIC-IV samples, bar one. Such a property might be a result of the way we selected the estimators. When comparing models trained on different train/test splits, we only looked at the Qini AUC score on the test set, which evaluates the quality of the ranking of samples ordered by the estimated CATE, but does not ensure that these estimates indeed match reality.

A big constraint of our evaluation on RCT data is the fact that three features from the MIMIC-IV dataset had to be omitted – *bilirubin*, *platelets* and *urea*. The last two, as mentioned in subsection 5.2, were identified as having a high impact on predictions. Without them, the models have most likely lost a part of their accuracy and predictive power. Furthermore, due to time constraints, this evaluation was performed only once. Broad confidence intervals for results of tests on MIMIC-IV data suggest that the models' performance was highly dependent on the used train/test split. Therefore, to fully measure the models' efficiency on the RCT data, the evaluation would need to be run multiple times with models trained each time on a different data split.

5.4 Limitations

In subsection 2.4 we have described which variables from the dataset have been selected based on literature, our talks with the supervisors of this paper and data-driven methods. However, we are not medical professionals and could have added unnecessary features or missed important ones. Therefore,

[2] For Shapley values we have used Python library *shap* (https://shap.readthedocs.io)

this list could be incorrect, introducing bias into all of the results we have obtained. An expert, experienced in the PEEP assignment process, should check that list for correctness before any of our results are used in practice.

Another limitation is connected to the causal inference assumptions. As mentioned in subsection 2.3 they are quite strong and are not guaranteed to hold in observational datasets, such as MIMIC-IV. For example, conditional exchangeability means that there is no unmeasured confounding in the data. However, Calfee et al. [25] managed to split patients into two groups with different mortality rates in response to a given PEEP level. The main splitting conditions were three features: interleukin-6, soluble tumour necrosis factor receptor-1 and vasopressor use [yes or no]. Following our reasoning about *pf_ratio* in subsection 2.2, one can say that these variables are confounders. However, they were not available in the MIMIC-IV data, thus our models could not account for them, potentially introducing bias into our results.

When one looks at the plots showing the evolution of MSE depending on the size of the training set in our simulation experiments (Figure 1 and Appendix A), it can be seen that the models begin to converge only at around 5000-10000 training samples. This result was roughly the same even when we ran the simulations without introducing any noise in the data. Therefore, our pre-processed MIMIC-IV dataset, which has only 3941 samples, might simply be too small to provide a large enough training sample for the models. Increasing the amount of data, by including other suitable datasets might improve the models and their CATE estimates.

6 Responsible Research

This paper contributes to the goal of understanding the PEEP assignment problem, which could potentially save lives in the future. Because of that, we have to reflect on the ethical aspects and reproducibility of our study.

6.1 MIMIC-IV dataset

Our research is heavily based on a database containing medical records of real patients admitted to the ICU of Beth Israel Deaconess Medical Center. Because of that, we need to pay special attention to preserving the anonymity of people whose data we use. Authors of the MIMIC-IV dataset have deidentified the data by taking several actions [10], for example: patient identifiers were replaced with random integer identifiers, a special algorithm was used to erase any protected health information (PHI) from free-text fields and a random offset was applied to all dates (while preserving their relative order for a given patient). Throughout our study we have not made any attempts at reidentifing the patient data. Moreover, to gain access to the database we had to complete a data privacy training – *CITI Data or Specimens Only Research* [3] course.

6.2 Potential bias

As a result of our pre-processing and variable selection steps we have not included features that have been historically discriminated against, such as race or gender, unless they have shown to be important in the models (e.g. age). However, other variables could act as proxies, for example it was shown that platelet levels differ among races [26] and bilirubin measures vary between sexes [27]. Because of that, it is possible that the models could discriminate against particular groups of people, and this would need to be be further investigated before the models are used in practice in any way.

6.3 Reproducibility

By extracting the main goals of the FAIR principles (designed for reusability of scholarly data) [28] and applying them to this paper and the codebase used for our experiments, we hope to make our research reproducible and allow others to independently verify our results.
Findable – The paper is freely accessible at the TU Delft repository and the code is publicly available online[4].
Accessible – Both paper and code do not require any special authorisation to be accessed and can be found by anyone with a connection to the Internet.
Interoperable – The code was written in Python, meaning it can be used on almost all modern-day computers.
Reusable – The codebase is well-documented and ready to be run again without any specific setup. Moreover, it is customisable, allowing others to carry out the experiments with different parameters.

7 Conclusions and Future Work

In this study, our goal was to investigate the performance of S-learner, T-learner (with random forest as the base model for these two estimators) and causal forest in the task of predicting the conditional average treatment effect for PEEP assignment. Through a series of experiments on simulated data, the MIMIC-IV dataset and samples from a randomised controlled trial, we measured the models' efficacy on specific use cases and determined whether they make valuable predictions in the goal of CATE estimation.

From the experiments on simulated data, we found that S-learner performed best where there was no treatment effect, T-learner outperformed other models when the response functions for treated and untreated groups were independent of each other, and causal forest performed overall well, achieving low error rates in almost all tests.

When trained on the MIMIC-IV data, the models were heavily overfitting to the training set and performed rather poorly. Hyperparameter tuning helped in fixing these issues, improving the models' performance, but did not alleviate them completely. Nonetheless, when trained on a correct train/test split, the models have showcased, to some extent, the ability to expose the heterogeneity in the effect of PEEP level assignment, distinguishing groups of patients that benefit or suffer from high PEEP, with T-learner noticeably outperforming the other two models. However, when evaluated on the RCT dataset, the models performed only marginally better than a theoretical model making random CATE predictions, although this finding could be a result of the fact that not all variables from the MIMIC-IV database were available in this dataset.

[3] https://about.citiprogram.org/

[4] https://github.com/Hubert1913/research-project

In section 5 we have highlighted several concerns and limitations we have encountered in this study. One could attempt to address some of them in further research. Including in our dataset more samples and more variables that affect treatment outcome, as well as running the experiments for more iterations could improve models' performance and reduce the width of confidence intervals in the results. Moreover, the evaluation on the RCT data could be performed multiple times, each time on models trained on a different part of the MIMIC-IV database, increasing the reliability of conclusions made from this test.

Lastly, we found that four features: *age*, *platelets*, *urea* and *pco2* had a strong influence on the CATE estimates made by all three models. These findings could be checked in future work, by for example performing clinical trials or inspecting if other models for estimating CATE also show this behaviour.

References

[1] National Heart, Lung, and Blood Institute ARDS Clinical Trials Network, "Higher versus lower positive end-expiratory pressures in patients with the acute respiratory distress syndrome," *New England Journal of Medicine*, vol. 351, no. 4, pp. 327–336, 2004.

[2] A. B. Cavalcanti, É. A. Suzumura, L. N. Laranjeira, D. de Moraes Paisani, L. P. Damiani, H. P. Guimarães, E. R. Romano, M. de Moraes Regenga, L. N. T. Taniguchi, C. Teixeira, *et al.*, "Effect of lung recruitment and titrated positive end-expiratory pressure (peep) vs low peep on mortality in patients with acute respiratory distress syndrome: a randomized clinical trial," *Jama*, vol. 318, no. 14, pp. 1335–1345, 2017.

[3] D. Ashbaugh, D. B. Bigelow, T. Petty, and B. Levine, "Acute respiratory distress in adults," *The Lancet*, vol. 290, no. 7511, pp. 319–323, 1967.

[4] S. K. Sahetya and R. G. Brower, "Lung Recruitment and Titrated PEEP in Moderate to Severe ARDS: Is the Door Closing on the Open Lung?," *JAMA*, vol. 318, pp. 1327–1329, 10 2017.

[5] M. Briel, M. Meade, A. Mercat, R. G. Brower, D. Talmor, S. D. Walter, A. S. Slutsky, E. Pullenayegum, Q. Zhou, D. Cook, L. Brochard, J.-C. M. Richard, F. Lamontagne, N. Bhatnagar, T. E. Stewart, and G. Guyatt, "Higher vs Lower Positive End-Expiratory Pressure in Patients With Acute Lung Injury and Acute Respiratory Distress Syndrome: Systematic Review and Meta-analysis," *JAMA*, vol. 303, pp. 865–873, 03 2010.

[6] A. Walkey, L. Sorbo, C. Hodgson, N. Adhikari, H. Wunsch, M. Meade, E. Uleryk, D. Hess, D. Talmor, B. Taylor, R. Brower, and E. Fan, "Higher peep versus lower peep strategies for patients with acute respiratory distress syndrome: A systematic review and meta-analysis," *Annals of the American Thoracic Society*, vol. 14, 10 2017.

[7] Y. Oba, D. M. Thameem, and T. Zaza, "High levels of peep may improve survival in acute respiratory distress syndrome: A meta-analysis," *Respiratory Medicine*, vol. 103, no. 8, pp. 1174–1181, 2009.

[8] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1228–1242, 2018.

[9] S. R. Künzel, J. S. Sekhon, P. J. Bickel, and B. Yu, "Metalearners for estimating heterogeneous treatment effects using machine learning," *Proceedings of the national academy of sciences*, vol. 116, no. 10, pp. 4156–4165, 2019.

[10] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "Mimic-iv (version 2.2)." PhysioNet (2023), https://doi.org/10.13026/6mm1-ek67.

[11] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow, *et al.*, "Mimic-iv, a freely accessible electronic health record dataset," *Scientific data*, vol. 10, no. 1, p. 1, 2023.

[12] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[13] C. Strodthoff, I. Frerichs, N. Weiler, and B. Bergh, "Predicting and simulating effects of peep changes with machine learning," *medRxiv*, pp. 2021–01, 2021.

[14] C. Händel, I. Frerichs, N. Weiler, and B. Bergh, "Prediction and simulation of peep setting effects with machine learning models," *Medicina Intensiva (English Edition)*, vol. 48, no. 4, pp. 191–199, 2024.

[15] A. Peine, A. Hallawa, J. Bickenbach, G. Dartmann, L. B. Fazlic, A. Schmeink, G. Ascheid, C. Thiemermann, A. Schuppert, R. Kindle, *et al.*, "Development and validation of a reinforcement learning algorithm to dynamically optimize mechanical ventilation in critical care," *NPJ digital medicine*, vol. 4, no. 1, p. 32, 2021.

[16] M. A. Pourhoseingholi, A. R. Baghestani, and M. Vahedi, "How to control confounding effects by statistical analysis," *Gastroenterology and hepatology from bed to bench*, vol. 5, no. 2, p. 79, 2012.

[17] M. A. Hernán and J. M. Robins, *Causal Inference: What If.* Boca Raton: Chapman & Hall/CRC, 2020.

[18] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[19] P. Bühlmann and B. Yu, "Analyzing bagging," *The annals of Statistics*, vol. 30, no. 4, pp. 927–961, 2002.

[20] N. Kiriakidou and C. Diou, "An evaluation framework for comparing causal inference models," in *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, SETN '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[21] F. Devriendt, J. Van Belle, T. Guns, and W. Verbeke, “Learning to rank for uplift modeling,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4888–4904, 2020.

[22] N. Radcliffe, “Using control groups to target on predicted lift: Building and assessing uplift model,” *Direct Marketing Analytics Journal*, pp. 14–21, 2007.

[23] S. Athey, J. Tibshirani, and S. Wager, “Generalized random forests,” *The Annals of Statistics*, vol. 47, no. 2, pp. 1148 – 1178, 2019.

[24] L. Shapley, *7. A Value for n-Person Games. Contributions to the Theory of Games II (1953) 307-317.*, pp. 69–79. Princeton: Princeton University Press, 1997.

[25] C. S. Calfee, K. Delucchi, P. E. Parsons, B. T. Thompson, L. B. Ware, and M. A. Matthay, “Subphenotypes in acute respiratory distress syndrome: latent class analysis of data from two randomised controlled trials,” *The Lancet Respiratory Medicine*, vol. 2, no. 8, pp. 611–620, 2014.

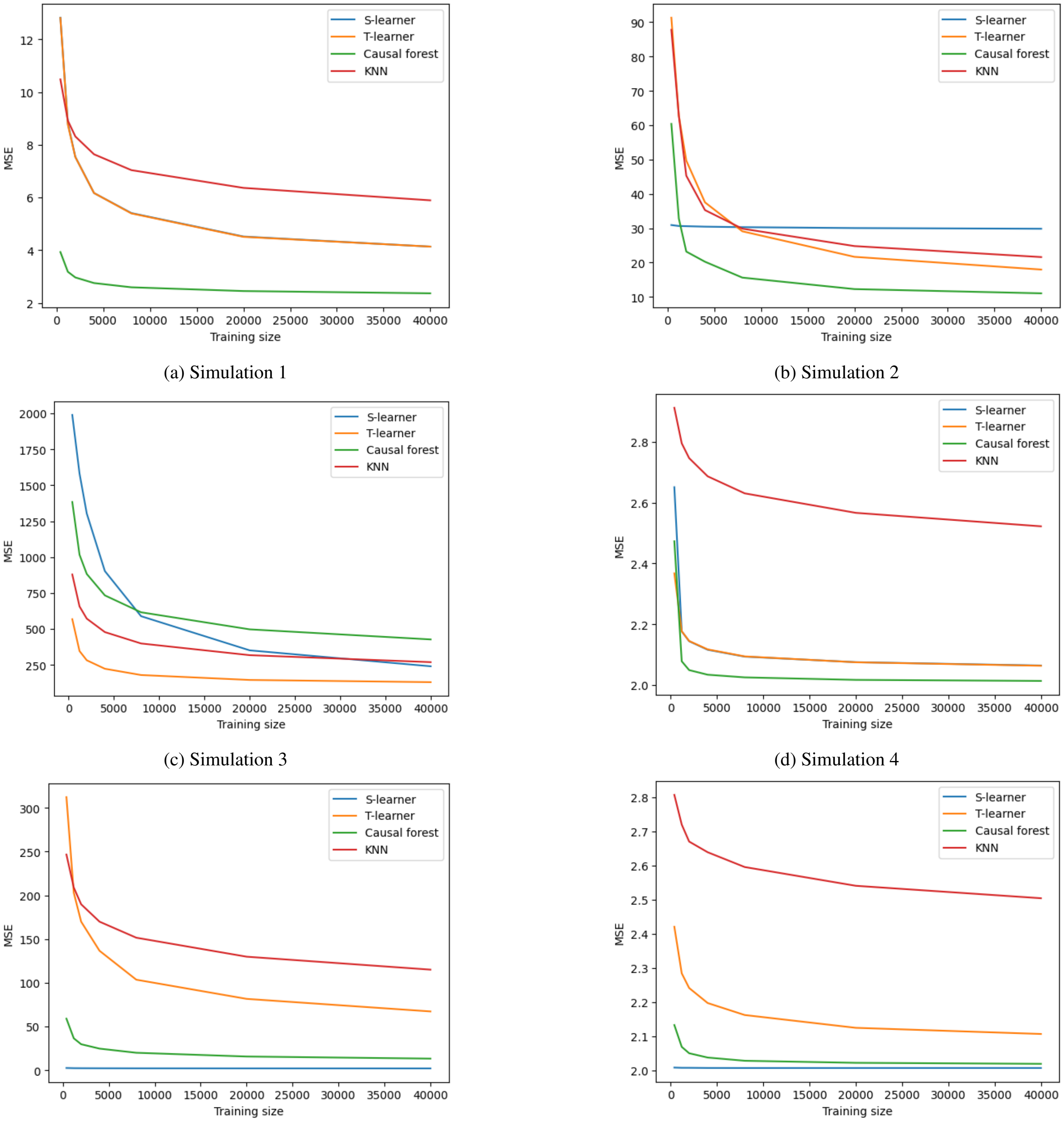
[26] B. J. Bain, “Ethnic and sex differences in the total and differential white cell count and platelet count.,” *Journal of clinical pathology*, vol. 49, no. 8, pp. 664–666, 1996.

[27] E. Lim, J. Miyamura, and J. J. Chen, “Racial/ethnic-specific reference intervals for common laboratory tests: a comparison among asians, blacks, hispanics, and white,” *Hawai'i Journal of Medicine & Public Health*, vol. 74, no. 9, p. 302, 2015.

[28] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons, “The fair guiding principles for scientific data management and stewardship,” *Scientific Data*, vol. 3, p. 160018, Mar 2016.

A Simulation MSEs

Figure 3 depicts the mean squared error between the real and estimated treatment effect on the test size as a function of the size of the training set. The detailed setups of each of the simulations can be found in Table 1. Each test was repeated 50 times and the plots show average MSE for a given training size.



(a) Simulation 1

(b) Simulation 2

(c) Simulation 3

(d) Simulation 4

(e) Simulation 5

(f) Simulation 6

Figure 3: Evolution of MSE in the simulation experiments

B Optimal hyperparameters values

Table 6 and Table 7 show the best hyperparameter values found using a grid search as described in subsection 4.2, respectively in cases when the models were trained on all features and only on four of them - *age*, *platelets*, *urea* and *pco2*.

Table 6: Optimal hyperparameters values for each of the models, using all features

Model	max_depth	min_samples_split	n_estimators	max_samples
S-learner	5	5	500	0.5
T-learner	3	5	250	0.5
Causal forest	5	10	2500	0.45

Table 7: Optimal hyperparameters values for each of the models, using only *age*, *platelets*, *urea* and *pco2*

Model	max_depth	min_samples_split	n_estimators	max_samples
S-learner	*None*	35	50	0.5
T-learner	20	50	250	0.5
Causal forest	3	5	24	0.45

C Qini curves for RCT data

Figure 4 presents the Qini curves for the treatment effect predictions the models made on the data from a randomised controlled trial.

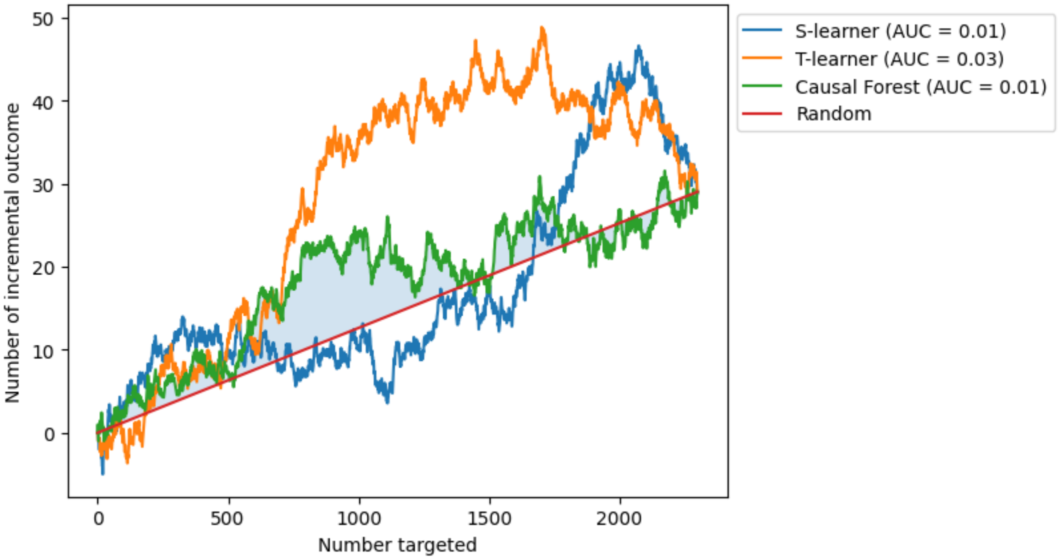
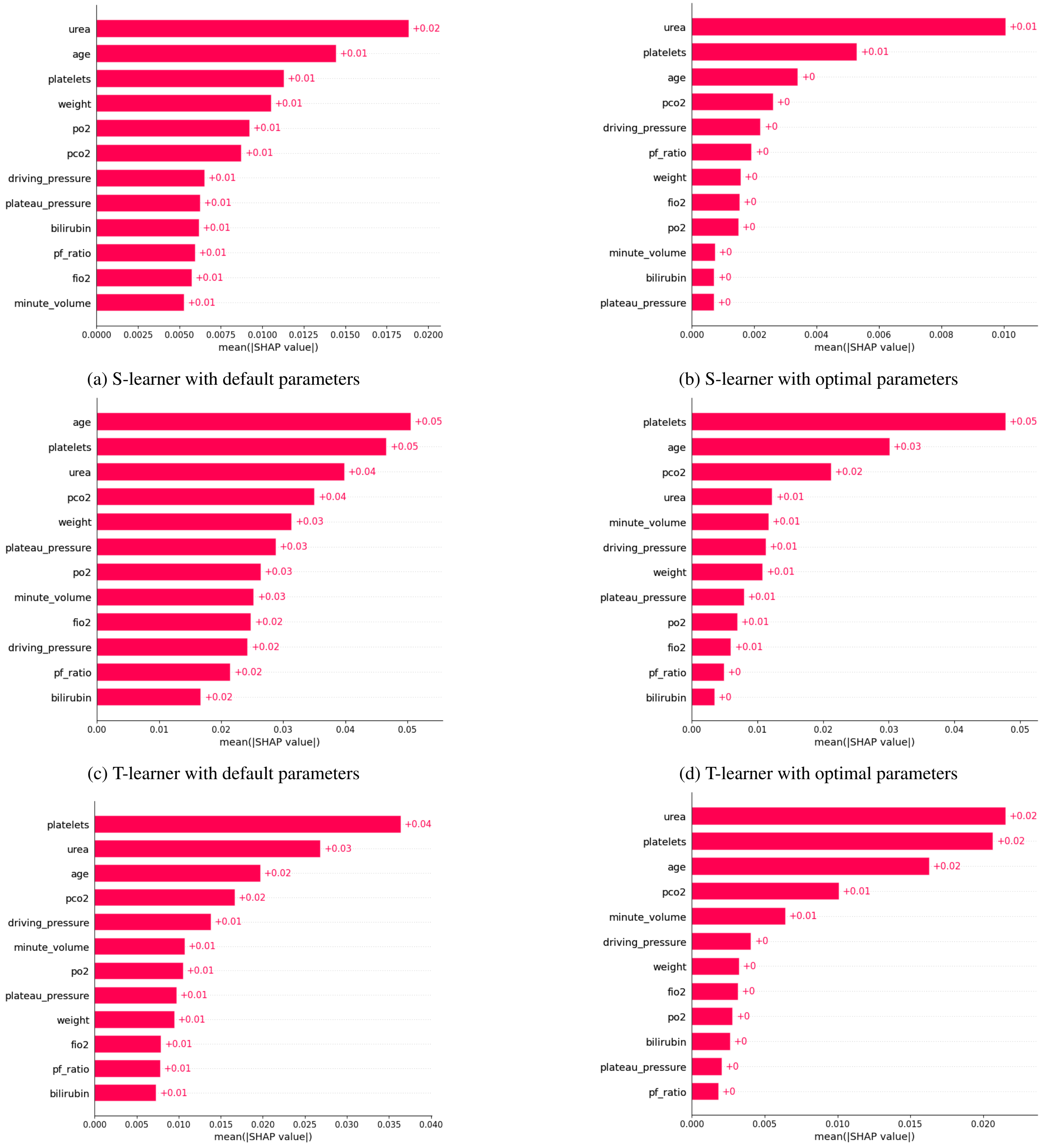


Figure 4: Qini curves for the predictions on RCT dataset

D Shapley values

Figure 5 depicts the average modulus of Shapley values of S-learner, T-learner and causal forest when trained on the MIMIC-IV dataset with default and optimal hyperparameters.



(a) S-learner with default parameters

(b) S-learner with optimal parameters

(c) T-learner with default parameters

(d) T-learner with optimal parameters

(e) Causal forest with default parameters

(f) Causal forest with optimal parameters

Figure 5: Average modulus of Shapley values of S-learner, T-learner and causal forest with default and optimal hyperparameters