Supporting information Appendix for Pichler & Hartig – Can machine learning be used for causal inference?

Table of contents

**Summary:** This document provides supporting information on Pichler & Hartig – Can machine learning be used for causal inference.

## 1 Extending MCE to two-way interactions

MCE can be extended to $n$-dimensions to detect way feature interactions. Here, we extended MCEs to two dimensions to detect two-way feature interactions by asking what the change is of when features and change together:

We can approximate with the finite difference method:

and are set to and . All features are centered and standardized.

## 2 Hyper-parameter tuning

We performed a hyper-parameter search to check if hyper-parameters can influence the unbiasness of the models. For that, we created simulation scenarios 100 and 1,000 observations, and each with 10 and 100 features. In each simulation scenario, two features (X1 and X5) had an effect on Y and all other features had no effect on Y. Features were sampled from a multivariate normal distribution and all features were randomly correlated (Variance-covariance matrix was sampled from a Wishart distribution (degrees of freedom = number of features (10 or 100), scale matrix was the identity matrix).

3,000 combinations of hyper-parameters were randomly drawn (Table S1). For each draw, the data was sampled, and bias for X1, X2, and X5 were calculated (bias = true effect - estimated effect, effects were estimated by MCE). In each draw, holdout data of the same size as the training data were used used to calculate the root mean squared error (RMSE) of the models.

**Table S** 1: Overview over hyper-parameters for Neural Network, Boosted Regression Tree, and Random Forest

| Algorithm | Hyper-parameter | Range |
| --- | --- | --- |
| Neural Network | activation function | [relu, leaky\_relu, tanh, selu, elu, celu, gelu] |
|  | depth | [1, 20] |
|  | width | [2, 171] |
|  | batch size (sgd) | [1, 100] |
|  | dropout rate | [0, 0.3] |
| Boosted Regression Tree | eta | [0.01, 0.4] |
|  | max depth | [2, 25] |
|  | subsample | [0.5, 1] |
|  | max tree | [30, 125] |
|  | lambda | [1, 20] |
| Random Forest | mtry | [2, 99] |
|  | min node size | [2, 70] |
|  | max depth | [2, 50] |
|  | regularization factor | [0, 1] |

### 2.1 Neural Network

|  |
| --- |
| **Figure S** 1: Trade-off between bias (feature X1, true effect = 1) and RMSE of the fittend NNs. Each point is a simulated scenario with a different set of hyper-parameters. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (activation functions, depth and width of the hidden layers, batch\_size (sgd), and dropout) were randomly sampled. |

For our data-poor simulation experiments with n p, we found only a weak correlation between the bias of X1 and the RMSE on the holdout data (Fig. S1). This correlation was slightly stronger in the data-poor situations (N = 100, P = 100, and N = 1000, P = 100; Fig. S1).

|  |
| --- |
| **Figure S** 2: Effects of hyper-parameters of neural networks on bias of feature X1. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (activation functions, depth and width of the hidden layers, batch\_size (sgd), and dropout) were randomly sampled. Effects of hyper-parameters were estimated by linear regression models. |

We found significant differences between the activation functions for the bias of X (true effect = 1) and the RMSE (N = 100, P = 100). Here, the ‘selu’ activation function had the smallest biases and RMSE (Fig. S2). With more data, or less features, the differences between the activation functions got smaller and for data-rich situations (N = 1000, P = 10), the confidence intervals of the different activation functions were in the same range (Fig. S2).

Depth had a strong effect on the bias and the RMSE, and this effect was stronger for the data-poor scenario (N = 100, P = 100; Fig. S2). Lower depth had smaller biases and lower RMSE. Width had a weak effect in the data-poor situation, but stronger effects in data-rich scenarios on the bias. But width had strong effects on the RMSE, increasing the width led to lower RMSE (this effect was weaker for the data-rich scenarios, Fig. S2). Dropout had only small effects on the biases and the RMSE.

Hyper-parameters had only small effects on the bias of the zero-effect (X2), thus it seems that the NN can identify, reliable, zero-effects independent of the hyper-parameters (Fig. S2).

### 2.2 Boosted Regression Trees

The correlation between RMSE and bias was stronger for data-rich scenarios than for the data-poor scenarios (Fig. S3). We found a stronger relation between RMSE and bias of X1 for BRT than for NN (Fig. S1, S3).

|  |
| --- |
| **Figure S** 3: Trade-off between bias (feature X1, true effect = 1) and RMSE of the fittend BRTs. Each point is a simulated scenario with a different set of hyper-parameters (3,000 samples). Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled. |

Overall, the effects on bias (X1 ) and RMSE were similar, although small (Fig. S4). Lambda and max tree had the largest effects on the biases and the RMSE (Fig. S3). Effects on the bias of the zero-effect, X2, were small and had large confidence intervals (Fig. S3).

|  |
| --- |
| **Figure S** 4: Effects of hyper-parameters of BRT on bias of feature X1. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled. Effects of hyper-parameters were estimated by linear regression models. |

#### 2.2.1 Confounder unequal scenario

As we found large biases in the proof-of-concept simulations for BRT, we repeated the hyper-paramter tuning for the worst scenario (unequal confounder, Fig. 2C in the main text).

We used the second best set (eta = 0.31, max depth = 5, subsample = 0.55, lambda = 4.29, max tree = 116) for the rest of analyses. because it had the best trade-off for the biases of three features and the RMSE (Table S2).

**Table** **:** Trade-off between bias (feature X~1~, true effect = 1) and RMSE of the fittend BRT Each point is a simulated scenario with a different set of hyper-parameters. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X~1~ and X~5~) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled (3,000 draws).

| **eta** | **max\_depth** | **subsample** | **lambda** | **max\_tree** | **Bias X1** | **Bias X2** | **Bias X3** | **rmse** | **weighted\_bias** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.267 | 4 | 0.645 | 3.904 | 116 | -0.002 | -0.036 | -0.007 | 0.363 | 0.016 |
| 0.381 | 2 | 0.590 | 5.346 | 112 | 0.029 | -0.062 | 0.007 | 0.349 | 0.043 |
| 0.300 | 3 | 0.544 | 1.088 | 104 | -0.001 | -0.031 | -0.032 | 0.356 | 0.065 |
| 0.327 | 3 | 0.583 | 7.349 | 98 | 0.036 | 0.040 | -0.015 | 0.360 | 0.066 |
| 0.382 | 2 | 0.533 | 2.769 | 77 | 0.060 | 0.102 | -0.006 | 0.355 | 0.071 |
| 0.310 | 3 | 0.803 | 4.531 | 97 | 0.004 | -0.058 | -0.051 | 0.346 | 0.106 |

### 2.3 Random Forest

For our data-poor simulation experiments with n p, RF showed a stronger correlation between bias of X1 and the RMSE (Fig. S5) than NN and BRT.

|  |
| --- |
| **Figure S** 5: Trade-off between bias (feature X1, true effect = 1) and RMSE of the fittend RF. Each point is a simulated scenario with a different set of hyper-parameters. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled. |

Bias of X1 was mostly only affect by the mtry parameter (Fig. S6), which was within our expectations because it controls the spill-over effect for collinear features.

|  |
| --- |
| **Figure S** 6: Effects of hyper-parameters of neural networks on bias of feature X1. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled. Effects of hyper-parameters were estimated by linear regression models. |

#### 2.3.1 Confounder unequal scenario

As we found large biases in the proof-of-concept simulations for RF, we repeated the hyper-paramter tuning for the worst, the last, scenario (unequal confounder, Fig. 2C in the main text).

We tried the second best set of hyper-parameters in the proof-of-concept simulations but it actually led to larger biases in Fig. 2A and 2B and we decided to fall back to the default values.

**Table** **:** Trade-off between bias (feature X~1~, true effect = 1) and RMSE of the fittend RF Each point is a simulated scenario with a different set of hyper-parameters. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X~1~ and X~5~) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled.

| **mtry** | **min.node.size** | **max.depth** | **regularization.factor** | **bias\_1** | **Bias X1** | **Bias X2** | **Bias X3** | **weighted\_bias** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 9 | 38 | 0.301 | 0.214 | -0.046 | -0.232 | 0.370 | 0.677 |
| 4 | 7 | 43 | 0.043 | 0.229 | -0.087 | -0.256 | 0.378 | 0.741 |
| 4 | 3 | 11 | 0.220 | 0.265 | -0.048 | -0.241 | 0.385 | 0.746 |
| 5 | 2 | 48 | 0.249 | 0.273 | -0.025 | -0.243 | 0.372 | 0.760 |
| 3 | 7 | 25 | 0.008 | 0.273 | -0.095 | -0.247 | 0.365 | 0.767 |
| 5 | 2 | 50 | 0.314 | 0.256 | -0.065 | -0.261 | 0.362 | 0.777 |

### 2.4 Elastic net

For our data-poor simulation experiments with n p, RF showed a stronger correlation between bias of X1 and the RMSE (Fig. S5) than NN and BRT.

|  |
| --- |
| **Figure S** 7: Trade-off between bias (feature X1, true effect = 1) and RMSE of the fittend Elastic-net. Each point is a simulated scenario with a different set of hyper-parameters. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (alpha and lambda) were randomly sampled. |

Bias of X1 was mostly only affect by the mtry parameter (Fig. S6), which was within our expectations because it controls the spill-over effect for collinear features.

|  |
| --- |
| **Figure S** 8: Effects of hyper-parameters of Elastic-net on bias of feature X1. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X1 and X5) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (alpha and lambda) were randomly sampled. Effects of hyper-parameters were estimated by linear regression models. |

#### 2.4.1 Confounder unequal scenario

As we found large biases in the proof-of-concept simulations for elastic-net, we repeated the hyper-paramter tuning for the worst, the last, scenario (unequal confounder, Fig. 2C in the main text).

We tried the second best set of hyper-parameters in the proof-of-concept simulations but it actually led to larger biases in Fig. 2A and 2B and we decided to fall back to the default values.

**Table** **:** Trade-off between bias (feature X~1~, true effect = 1) and RMSE of the fittend ELastic-net Each point is a simulated scenario with a different set of hyper-parameters. Scenarios with 1,00 and 1,000 observations and with 100 or 10 features (P) were simulated. In each scenario, only two features (X~1~ and X~5~) had an effect (effect size = 1) on the response variable Y. Hyper-parameters (max\_depth, subsample, lambda, and max\_tree) were randomly sampled.

| **alpha** | **lambda** | **Bias X1** | **Bias X3** | **Bias X2** | **rmse** | **weighted\_bias** |
| --- | --- | --- | --- | --- | --- | --- |
| 0.829 | 0.001 | 0.006 | -0.001 | -0.001 | 0.312 | 0.007 |
| 0.294 | 0.001 | 0.003 | -0.003 | 0.004 | 0.290 | 0.012 |
| 0.058 | 0.003 | -0.002 | -0.028 | 0.012 | 0.301 | 0.026 |
| 0.072 | 0.005 | 0.014 | -0.013 | -0.009 | 0.305 | 0.032 |
| 0.841 | 0.001 | 0.013 | -0.028 | -0.012 | 0.301 | 0.037 |
| 0.477 | 0.001 | -0.027 | 0.019 | 0.014 | 0.302 | 0.054 |

### 2.5 Comparison of algorithms

|  | RF | BRT | NN | elastic-net |
| --- | --- | --- | --- | --- |
| R2 between RMSE and bias of X1 | 0.101 | 0.214 | 0.245 | 0.901 |
| RMSE tuned bias of X1 | -0.751 | -0.019 | -0.187 | -0.124 |
| RMSE tuned RMSE | 0.924 | 0.375 | 0.472 | 0.264 |
| Bias-tuned bias of X1 | -0.702 | -0.001 | -0.009 | 0.003 |
| Bias-tuned (X1 ) RMSE | 1.03 | 0.615 | 0.674 | 0.432 |

**Table S** 2: Selected hyper-parameterfor ML algorithms

| Algorithm |  |
| --- | --- |
| RF | mtry=0.00154429720714688, min.node.size=23, max.depth=10, regularization.factor=0.553951787063852 |
| BRT | eta=0.218754931651056, max\_depth=2, subsample=0.795404108474031, lambda=3.94214590545744, max\_tree=105 |
| NN | activations=selu, sgd=30, depth=2, width=22, dropout=0.0558242111932486 |
| elastic-net | alpha=0.958145375363529, lambda=0.0481374466326088 |

## 3 Proof of concept - Additional results

All models (Fig. 2) showed small variances (Fig. S7)

|  |
| --- |
| **Figure S** 9: Variances of effect estimates for different ML algorithms in three different simulated causal simulations (a, b, and c). Sample sizes are so large that stochastic effects can be excluded. Effects of the ML models were inferred using marginal conditional effects. Row a) shows results for simulations with uncorrelated features with effect sizes (x1: 1.0, x2: 0.5, x3: 1.0). Row b) shows the results for simulations with x1 and x2 being strongly correlated (Pearson correlation factor = 0.9) but only x1 has an effect on y (mediator) and row c) shows the results for x1 and x2 being strongly correlated (Pearson correlation factor = 0.9 with x1 and x2 having effects on y (confounder scenario) |

### 3.1 NN with Dropout, LASSO, and Ridge

We additionally tested NN with dropout (rate = 0.2), LASSO regression, and Ridge regression for the proof-of-concept simulations.

|  |
| --- |
| **Figure S** 10: Bias on effect estimates for additional ML algorithms (NN with Dropout, LASSO, and Ridge regression) in three different simulated causal simulations (a, b, and c).Sample sizes are so large that stochastic effects can be excluded. Effects of the ML models were inferred using marginal conditional effects. Row a) shows results for simulations with uncorrelated features with effect sizes (x1: 1.0, x2: 0.5, x3: 1.0). Row b) shows the results for simulations with x1 and x2 being strongly correlated (Pearson correlation factor = 0.9) but only x1 has an effect on y (mediator) and row c) shows the results for x1 and x2 being strongly correlated (Pearson correlation factor = 0.9 with x1 and x2 having effects on y (confounder scenario) |

All three methods showed biased estimates in the first scenario for effects without collinearity (Fig. S8). With collinearity, all three showed larger biases but Ridge regression showed the largest biases (Fig. S8).

|  |
| --- |
| **Figure S** 11: Variances of effect estimates for additional ML algorithms (NN with Dropout, LASSO, and Ridge regression) in three different simulated causal simulations (a, b, and c). Sample sizes are so large that stochastic effects can be excluded. Effects of the ML models were inferred using marginal conditional effects. Row a) shows results for simulations with uncorrelated features with effect sizes (x1: 1.0, x2: 0.5, x3: 1.0). Row b) shows the results for simulations with x1 and x2 being strongly correlated (Pearson correlation factor = 0.9) but only x1 has an effect on y (mediator) and row c) shows the results for x1 and x2 being strongly correlated (Pearson correlation factor = 0.9 with x1 and x2 having effects on y (confounder scenario) |

While estimates from NN with dropout had the smallest biases, they had the largest variances (Fig. S9), though these variances were still small (Fig. S9).

### 3.2 RMSE on holdout

In addition to bias and variance, we calculated the predictive error using the RMSE on holdout data which had the same size as the training data (N = 1000).

RF, BRT, and Dropout showed the highest RMSE in all three scenarios (Fig. S10). LASSO and elastic-net showed the smallest RMSE in all three scenarios (Fig. S10).

|  |
| --- |
| **Figure S** 12: Root mean squared error (RMSE) for different ML algorithms in three different simulated causal simulations (a, b, and c). Sample sizes are so large that stochastic effects can be excluded. 1,000 observations were used to train the models and 1,000 observations were used to evaluate the predictive performance of the models. Column ‘effects’ shows results for simulations with uncorrelated features with effect sizes (x1: 1.0, x2: 0.5, x3: 1.0). Column ‘collinearity\_0.90’ shows the results for simulations with x1 and x2 being strongly correlated (Pearson correlation factor = 0.9) but only x1 has an effect on y (mediator) and column ‘confounder\_unequal’ shows the results for x1 and x2 being strongly correlated (Pearson correlation factor = 0.9 with x1 and x2 having effects on y (confounder scenario). |

## 4 Data-poor simulation

### 4.1 Without feature interactions

We additionally calculated the RMSE on holdout-data of the same size as the training data for the data-poor simulations (Fig. 3). For N = 100, LM showed the highest RMSE, which is explained by the large biases and variances (Fig. 3). Overall, RF and Ridge regression showed the highest RMSE (Fig. S11).

|  |
| --- |
| **Figure S** 13: Root mean squared error (RMSE) of models for data-poor simulations. N = 100 (low), 600 (med), 2,000 (large) was simulated with 100 features. All features were weakly correlated. Two features had main effects (effect sizes = 1, X1 and X5) on the response and the other 95 features had no effects. Models (LM, RF, BRT, NN, Elastic-net) were fitted on the simulated data and holdout data of the same size as the training data was used to estimate predictive error (RMSE). Simulation and model fitting were repeated 100 times. |

### 4.2 With feature interactions and collinearity

Moreover, we simulated scenarios with only 29 features but with interactions between features. We sampled the 29 features from a multivariate normal distribution (mean vector ) and the covariance matrix was sampled from a Wishart distribution (degress of freedom = 29) to create a weak collinear structure between all 29 features.

The data-generating model was with . All other features had no effects on the outcome.

We sampled 100, 600, and 2,000 observations from the data-generating model. Holdout-data of the same size was used to calculate the RMSE of the model. Simulation and model fitting was repeated 100 times.

For the main effects, we found similar results as for the main effects from Fig. 3, for the interactions, we found that the biases were higher than as for the main effects (Fig. S14) but it also decreased with the number of observations. Here, elastic-net showed the best performance, however, only elastic-net and LM received all combinatorical possible feature interactions as features where as RF, BRT, and NN had to infer the interactions automatically from the features. Again, BRT and NN showed significant lower biases than RF.

|  |
| --- |
| **Figure S** 14: Bias and variance of estimated effects in a data-poor situation. N = 100, 600, 2,000 was simulated with 29 features. All features were weakly correlated. Two features had main effects (effect sizes = 1) on the response and 4 features had two interaction effects (effect sizes = 1) on the response. Models (LM, RF, BRT, NN, Elastic-net) were fitted on the simulated data and effect estimates were approximated using MCEs. LM and elastic-net received all possible two-way interactions (406) as features. Simulation and model fitting were repeated 100 times. For all effects (Main = 1, Inter = 1) and non-effects (Main = 0, Inter = 0), bias and variance of the effect estimates were calculated. |

|  |
| --- |
| **Figure S** 15: Root mean squared error (RMSE) of models for data-poor simulations. N = 100 (low), 600 (med), 2,000 (large) was simulated with 29 features. All features were weakly correlated. Two features had main effects (effect sizes = 1, X1 and X5) and there were two interaction effects (X1:X2 and X3:X4) on the response and the other 95 features had no effects. Models (LM, RF, BRT, NN, Elastic-net) were fitted on the simulated data and holdout data of the same size as the training data was used to estimate predictive error (RMSE). LM and elastic-net received all possible two-way interactions (406) as features. Simulation and model fitting were repeated 100 times. |

### 4.3 With feature interactions and without collinearity

We repeated the simulation with feature interactions but without collinearity, i.e.  of the multivariate normal distribution, from we sampled the 29 features, was the identity matrix.

We found similar results as before, i) bias was larger for the feature interactions for RF, BRT, and NN as on the main effects, ii) biases decreased with number of observations, and iii) RF showed the largest biases for 2,000 observations (Fig. S14).

|  |
| --- |
| **Figure S** 16: Bias and variance of estimated effects in a data-poor situation. N = 100, 600, 2,000 was simulated with 29 features. All features were uncorrelated. Two features had main effects (effect sizes = 1) on the response and 4 features had two interaction effects (effect sizes = 1) on the response. Models (LM, RF, BRT, NN, Elastic-net) were fitted on the simulated data and effect estimates were approximated using MCEs. LM and elastic-net received all possible two-way interactions (406) as features. Simulation and model fitting were repeated 100 times. For all effects (Main = 1, Inter = 1) and non-effects (Main = 0, Inter = 0), bias and variance of the effect estimates were calculated. |

For 100 observations (low), again, LM should the highest RMSE which is not surprising given the large biases and variances (Fig. S14). RF, BRT, and NN with dropout showed the largest RMSE, whereas elastic-net, LASSO, Ridge showed the lowest RMSE (Fig. S15).

|  |
| --- |
| **Figure S** 17: Root mean squared error (RMSE) of models for data-poor simulations. N = 100 (low), 600 (med), 2,000 (large) was simulated with 29 features. All features were uncorrelated. Two features had main effects (effect sizes = 1, X1 and X5) and there were two interaction effects (X1:X2 and X3:X4) on the response and the other 95 features had no effects. Models (LM, RF, BRT, NN, Elastic-net) were fitted on the simulated data and holdout data of the same size as the training data was used to estimate predictive error (RMSE). LM and elastic-net received all possible two-way interactions (406) as features. Simulation and model fitting were repeated 100 times. |

## 5 Weighted MCE

If the instances of a feature x\_j are not uniformly distributed, we propose to calculate a weighted with the being, for example, the inverse probabilities of an estimated density function over the feature space of .

To demonstrate the idea of weighted MCE, we simulated a scenario with one feature where the for values of the feature and for the other feature values (Fig. S4). The feature was sampled from a log-Normal distribution. We fitted a linear regression model and a NN on the data and compared the effect estimated by the LM, the unweighted MCE, and the weighted MCE.

The LM estimated an effect of 1.48, the unweighted MCE was 1.95, and the weighted MCE was 1.48 (Fig. S16).

|  |
| --- |
| **Figure S** 18: Simulation example with non-uniform sampled feature X1 (log normal distributed). The red line is the effect estimated by a LM OLS. The blue line is the effect reported by an unweighted MCE from a NN. The green line is the effect reported by a weighted MCE from a NN. |

## 6 Case study - RMSE

**Table** **:** In-sample R2 of BRT, RF, NN, and LM in predicting the risk of Lung Cancer

| **name** | **BRT** | **RF** | **NN** | **LM** |
| --- | --- | --- | --- | --- |
| Causal ML | 0.58 | 0.54 | 0.63 | 0.64 |
| Conventional ML 1 | 0.84 | 0.84 | 0.87 | 0.87 |

**Table** **:** Out-of-sample R2 of BRT, RF, NN, and LM in predicting the risk of Lung Cancer with intervention on Lung Volume

| **name** | **BRT** | **RF** | **NN** | **LM** |
| --- | --- | --- | --- | --- |
| Causal ML | 0.58 | 0.54 | 0.64 | 0.64 |
| Conventional ML 1 | 0.45 | 0.48 | 0.41 | 0.41 |

**Table** **:** Out-of-sample R2 of BRT, RF, NN, and LM in predicting the risk of Lung Cancer with changed correlation structure because of the unobservable confounder Stress

| **name** | **BRT** | **RF** | **NN** | **LM** |
| --- | --- | --- | --- | --- |
| Conventional ML 1 | 0.60 | 0.55 | 0.67 | 0.68 |
| Causal ML | 0.85 | 0.73 | 0.94 | 0.97 |
| Conventional ML 2 | 0.89 | 0.75 | 0.97 | 0.99 |