GigaScience, 2023, **12**, 1–14
DOI: 10.1093/gigascience/giad071
Research

Confound-leakage: confound removal in machine learning leads to leakage

Sami Hamdan [1,2], Bradley C. Love [3,4,5], Georg G. von Polier [1,6,7], Susanne Weis [1,2], Holger Schwender [8], Simon B. Eickhoff [1,2] and Kaustubh R. Patil [1,2,*]

[1] Institute of Neuroscience and Medicine, Brain and Behaviour (INM-7), Forschungszentrum Jülich, 52428 Jülich, Germany
[2] Institute of Systems Neuroscience, Medical Faculty, Heinrich-Heine University Düsseldorf, 40225 Düsseldorf, Germany
[3] Department of Experimental Psychology, University College London, WC1H 0AP London, UK
[4] The Alan Turing Institute, London NW1 2DB, UK
[5] European Lab for Learning & Intelligent Systems (ELLIS), WC1E 6BT, London, UK
[6] Department of Child and Adolescent Psychiatry, Psychosomatics and Psychotherapy, University Hospital Frankfurt, 60528 Frankfurt, Germany
[7] Department of Child and Adolescent Psychiatry, Psychosomatics and Psychotherapy, RWTH Aachen University, 52074 Aachen, Germany
[8] Institute of Mathematics, Heinrich-Heine University Düsseldorf, 40225 Düsseldorf, Germany
*Correspondence address. Kaustubh R. Patil. Institute of Neuroscience and Medicine, Brain and Behaviour (INM-7), Forschungszentrum Jülich, Jülich, 52428 Germany. E-mail: k.patil@fz-juelich.de

Abstract

Background: Machine learning (ML) approaches are a crucial component of modern data analysis in many fields, including epidemiology and medicine. Nonlinear ML methods often achieve accurate predictions, for instance, in personalized medicine, as they are capable of modeling complex relationships between features and the target. Problematically, ML models and their predictions can be biased by confounding information present in the features. To remove this spurious signal, researchers often employ featurewise linear confound regression (CR). While this is considered a standard approach for dealing with confounding, possible pitfalls of using CR in ML pipelines are not fully understood.

Results: We provide new evidence that, contrary to general expectations, linear confound regression can increase the risk of confounding when combined with nonlinear ML approaches. Using a simple framework that uses the target as a confound, we show that information leaked via CR can increase null or moderate effects to near-perfect prediction. By shuffling the features, we provide evidence that this increase is indeed due to confound-leakage and not due to revealing of information. We then demonstrate the danger of confound-leakage in a real-world clinical application where the accuracy of predicting attention-deficit/hyperactivity disorder is overestimated using speech-derived features when using depression as a confound.

Conclusions: Mishandling or even amplifying confounding effects when building ML models due to confound-leakage, as shown, can lead to untrustworthy, biased, and unfair predictions. Our expose of the confound-leakage pitfall and provided guidelines for dealing with it can help create more robust and trustworthy ML models.

Keywords: confounding, data-leakage, machine-learning, clinical applications

Key Points:

- Confound removal is essential for building insightful and trustworthy machine learning (ML) models.
- Confound removal can increase performance when combined with nonlinear ML.
- This can be due to confound information leaking into the features.
- Possible reasons are skewed feature distributions and the feature of limited precision.
- Confound removal should be applied with utmost care in combination with nonlinear ML.

Introduction

Machine learning (ML) approaches have revolutionized biomedical data analysis by providing powerful tools, especially nonlinear models, that can model complex feature–target relationships [1, 2]. However, the very power these nonlinear models bring to data anar models, that can model complex feature–target realysis also leads to new challenges. Specifically, as we will detail, when a standard confound removal approach is paired with nonlinear models, new and surprising issues arise as the unintended is discovered and misinterpreted as a true effect.

Imagine building a diagnostic classifier for attention-deficit/hyperactivity disorder (ADHD) based on speech patterns. This will be a useful clinical tool aiding objective diagnosis [3]. However, like most disorders, ADHD has comorbidity, for instance, with depression. Ideally, an ADHD diagnostic classifier should only rely upon characteristics of ADHD and ignore that of depression. This is an example of confounding, where it is desirable that the confound depression is disregarded by the classifier. Another example of confounding is the effect of aging and neurodegenerative diseases on the brain. In a study to build a neuroimaging-based diagnostic classifier, the nonpathological

Received: January 6, 2023. **Revised:** June 1, 2023. **Accepted:** August 17, 2023

aging signal is confounding [4]. Confounding is ubiquitous, and further examples include batch effects in genomics [5–7], scanner effects in neuroimaging [8], patient and process information in radiographs [9], and group differences like naturally different brain sizes in investigation of brain size–independent sex differences [10, 11]. Ignoring confounding effects in an ML application can render predictions untrustworthy and insights questionable [12] as this information can be exploited by learning algorithms [13], leading to spurious feature–target relationships [14] (e.g., classification based on depression instead of ADHD or age instead of neuronal pathology). The benefits of big data in ML applications are obvious, especially when modeling weak relationships, but big data also lead to an increased risk of inducing confounded models [4, 11, 15, 16]. Confounding, thus, is a crucial concern and, if not properly, treated can threaten real-world applicability of ML.

When confounding masks the true feature–target relationship, its removal can clean the signal of interest, leading to higher generalizability (e.g., removal of batch effects in genomics) [7]. On the other hand, when confounding introduces artifactual relationships, the same procedure can reduce prediction accuracy [17, 18]. In either case, removing or adjusting for confounding effects is crucial for obtaining unbiased results, as otherwise an ML model might mostly rely on confounds, rendering signals of interest redundant. Two methods for treating confounding are commonly employed in data analysis with the goal of building an accurate ML model that is not biased by the confounding information. Data can be stratified based on the confounding variables, but it may introduce confounding information [19], falsely increase test-set performance by removing harder-to-classify data points [20], and can result in excessive data loss. As confounds share variation—usually presumed linear variance—with both the target and the features, another common method is confound regression (CR), which removes the confounding variance, also called confounded signal, from each feature separately using a linear regression model [4, 20]. The resulting residualized features are considered confound free and used for subsequent analysis. CR has become the default method to counter confounding in observational studies, including in ML applications [16, 20, 21]. Typically, a 2-step CR–ML workflow is constructed while avoiding risks associated with typical data leakage by applying CR in a cross-validation–consistent manner [20, 22]. It is important to note that we use a practitioner-oriented operational definition of confounds as a set of variables suspected to share an unwanted effect with both the features and target, which does not imply causality as in more formal definitions [23].

A CR–ML workflow typically attenuates prediction performance as it removes variance from the features that is informative of the target. If an increase in performance is observed after CR, it can be explained by either (i) *information-reveal*: CR reveals information that was masked by confounding or (ii) *confound-leakage*: leakage of confounding information into the features. In the case of information-reveal, CR could suppress linear confounding or noise, in turn enhancing the underlying (non)linear signal and making learning easier for a suitable ML algorithm [13]. This would be a positive effect similar to removing simple shortcuts in the data [24, 25]. If this is the case, then the resulting CR–ML workflow would be valuable for modeling nonlinear relationships. Alternatively, as CR is a univariate operation applied to each feature, multivariate confounding (across features) could be revealed, which could help prediction albeit undesirably. On the other hand, confound-leakage would be an even more worrisome outcome as it would leak confounding information into the features instead of removing it. Confound-leakage would be detrimental to the validity and interpretability of the ensuing CR–ML workflow and in some cases could lead to dangerous outcomes. CR has been reported to induce biases into statistical workflows, albeit not incorporating ML, leading to incorrectly inflated group differences inference in combined batch effects removal and group difference analysis [26]. It is important to note that CR is not without other pitfalls; for instance, it might fail to completely remove confounding information [21, 27]. Still, CR is considered the de facto method, and therefore analyzing the hitherto unknown pitfall of leaking confounding information through CR is helpful. Furthermore, there were speculations of confound-leakage in ML workflows [18], but it has not yet been systematically shown, analyzed, or explained.

To disentangle the 2 possible explanations of performance increase after CR, we systematically analyzed the 2-step CR–ML workflow. For analysis purposes and to gain detailed knowledge, we propose a framework that uses the target as a confound (TaCo), in which we use a single confound that is the target. As a confound needs to share variation with both the target and the feature, any possible confound must share all confounded signal with the target. Hence, the target can be seen as a "superconfound," subsuming all possible confounding effects. Although it is unlikely to encounter a confound equal to the target in real applications, TaCo provides a framework for systematic evaluation. It should be noted that real confounds will fall on the continuum from weak (low confounded signal) to strong (TaCo) depending on their degree of similarity with the target. Indeed, as we show, the TaCo framework reveals strong effects where the prediction accuracy is boosted from moderate to perfect as well as weaker effects for confounds weakly correlated with the target. A previous work has used TaCo for evaluating the validity and reliability of confound adjustment methods [21].

To this end, we performed extensive empirical analyses on several benchmark datasets, providing strong evidence for confound-leakage. First, we showcase confound-leakage in walk-through analyses. Then, using the TaCo framework, we systematically answer whether the improvement in prediction performance after CR is due to leakage. For this, we used benchmark datasets as well as several conceptually simple simulations covering both classification and regression problems. Finally, with a clinically relevant task of ADHD diagnosis using speech-related features with depression as a confound, we demonstrate the misleading impact of confound-leakage.

Results

Walk-through analysis

The goal of this section is to introduce readers to our analysis approach with intuitive examples. We show 1 exemplary case of TaCo removal for a binary classification task and a CR scenario with a weaker confound in a regression task. In both cases, we randomly split the data into 70% train and 30% test parts. The CR and prediction models were learned on the training data, and the results are reported on the test split. We will show that confound-leakage can be concluded if performance increases after performing CR on shuffled features ($\tilde{X}_{CR}$).

TaCo removal for binary classification

We analyzed the "bank investment" data to predict whether a customer will subscribe to term deposit given their financial and socioeconomic information. We used a decision tree (DT) with limited maximum depth of 2 for visualization ease. This example is meant to demonstrate key aspects of our proposed analyses (Fig. 1).

TaCo removal showed a much higher area under the curve for the receiver operating characteristic curve (AUCROC) of 0.98 compared to the baseline AUCROC of 0.75 without CR. Still, the TaCo-removed features were highly similar to the original features (median Pearson's correlation: 0.99, Fig. 1A,B). The 2 ensuing DTs were, however, completely different and relied on different features. Notably, these drastic differences were induced by minute feature alterations after CR that are hardly detectable by humans but are effectively captured by DT (Fig. 1C,D). Such performance increase can be due to revealed information or confound-leakage. Therefore, we sought to gain evidence to distinguish between these 2 scenarios using 2 complementary measurements: (i) destroying the relationship between features and target and (ii) use of confound-predicted features.

To destroy the feature–target relation, we shuffled each feature before CR ($\tilde{X}$) to create $\tilde{X}_{CR}$ and repeated the analysis. As there should be no predictive information in the shuffled features, the only explanation for above chance-level performance is CR leaking information into the confound-removed features X_{CR} (i.e., confound-leakage). We applied the shuffling procedure to a train-test split in this walk-through analysis. But it should be noted that when combined with a (nested) cross-validation and Bayesian Region of Practical Equivalence (ROPE) approach, this procedure can be used to compare models similarly as a permutation test (see section "Feature shuffling approach"). We observed chance-level performance without CR (AUCROC = 0.48) for the shuffled features. However, a performance increase after TaCo removal was observed (AUCROC = 0.99). This analysis shows that performance increase after TaCo removal with shuffled features indicates the possibility of confound-leakage.

Confound removal for regression

As an example of a weaker confound on a regression task, we simulated a binary confound and then sampled a feature from different distributions for each confound value (confound equal to 0 or 1). Then we added the confound to a normally distributed target ($M = 0$ and $SD = 0.50$; Fig. 1E,F). This creates a clear confounding situation, where the confound affects both the feature (point-biserial correlation = 0.71, $P < 0.01$) and the target (point-biserial correlation = 0.71, $P < 0.01$) and thus leads to a spurious relationship between the feature and the target (Pearson's correlation = 0.51, $P < 0.01$). Following the same procedure as in the previous example, we observed increased performance after CR using a DT with limited depth of 2 (R^2 using $X = 0.29$, $X_{CR} = 0.42$). As in these simulated data, only a spurious relation (via confound) exists between the feature and target, it is safe to assume that an increased performance after CR is due to confound-leakage. Furthermore, we found a probable mechanism behind this confound-leakage to be the distribution of the features conditioned on the confound. More precisely, CR shifts the feature values for confound = 1 in between most feature values for the confound = 0 (Fig. 1E). This leaks the confounding information into the feature instead of removing it (Fig. 1F). The shuffled features, however, were not sensitive to confound-leakage ($X = 0, \tilde{X} = -0.01$), which is expected considering the probable cause for such leakage depends on the joint distribution of the confound and the feature. When shuffling the features within each confound category to preserve the joint distribution, we observed an increase in performance after CR ($M = 0.29$ before to $M = 0.42$). This result indicates that shuffling the features might not be always sensitive to confound-leakage. We, nevertheless, use independently shuffled features in our analysis for practicality, particularly in the context of continuous or multiple confounding factors.

Analyses of benchmark data

TaCo removal increases performance of nonlinear methods

Our systematic and cross-validation (CV)–consistent analysis comprised comparison between TaCo removal pipelines and no-CR pipelines on 10 UC Irvine (UCI) datasets. TaCo removal led to a meaningful increase in out-of-sample scoring using all tested nonlinear models, random forest (RF) (7/10 datasets), DT (8/10), support vector machine (SVM) with radial basis function (RBF) kernel (5/10), and multilayer perceptron (MLP) (7/10) (Fig. 2, Supplementary Fig. S1). This suggests that confound-leakage is a risk associated with the usage of a CR–ML pipeline with nonlinear ML models. Furthermore, this suggests that the DT-based algorithms (DT and RF) are most susceptible to showing increased performance.

CR using weaker confounds also increases performance

As the target is the strongest possible confound, TaCo represents an extreme case. To test whether the potential leakage we found with TaCo extends to CR in general, using the UCI datasets, we simulated confounds related to the target at different strengths measured by Pearson's correlation ranging from 0.2 to 0.8. Depending on the dataset, different amounts of correlated confounds led to leakage after CR. We observed potential confound-leakage for 5 of the 10 datasets with at least 1 of the confound-target strengths. As expected, a higher target-confound correlation led to more leakage (i.e., higher performance after CR) (Fig. 2C).

Increased performance after TaCo removal is due to confound-leakage

As described in the walk-through analysis (see "TaCo removal for binary classification"), we measure the performance after first shuffling the features to evaluate whether the increased performance after TaCo removal/CR is due to information reveal or confound-leakage. After shuffling the features, both pipelines, no CR and TaCo removal, should perform close to chance level if the improved performance is due to revealed information. Indeed, the no-CR pipeline performed close to the chance level, while the TaCo-removal pipeline increased the performance (Fig. 2, TaCo CR Shuffled). As there should be no predictive information in the shuffled features, above chance-level performance could only be obtained if the CR leaks information. Thus, this result provides strong evidence in favor of the confound-leakage.

For the simulated weaker confounds, these results were less strong, but we still found 5 of 10 datasets where X_{CR} and 9 of 10 where $\tilde{X}_{CR}$ performed above chance level.

Possible mechanisms for confound-leakage

As a multitude of mechanisms could lead to confound-leakage, exhaustively identifying all possible mechanisms is out of the scope of this article. Rather, we want to highlight 2 possible mechanisms leading to confound-leakage inspired by the walk-through analyses: (i) confound-leakage due to continuous features deviating from normal distributions (see "Confound removal for regression") and (ii) confound-leakage due to unbalanced features of limited precision (see "TaCo removal for binary classification"). Both mechanisms could be summarized under the umbrella of

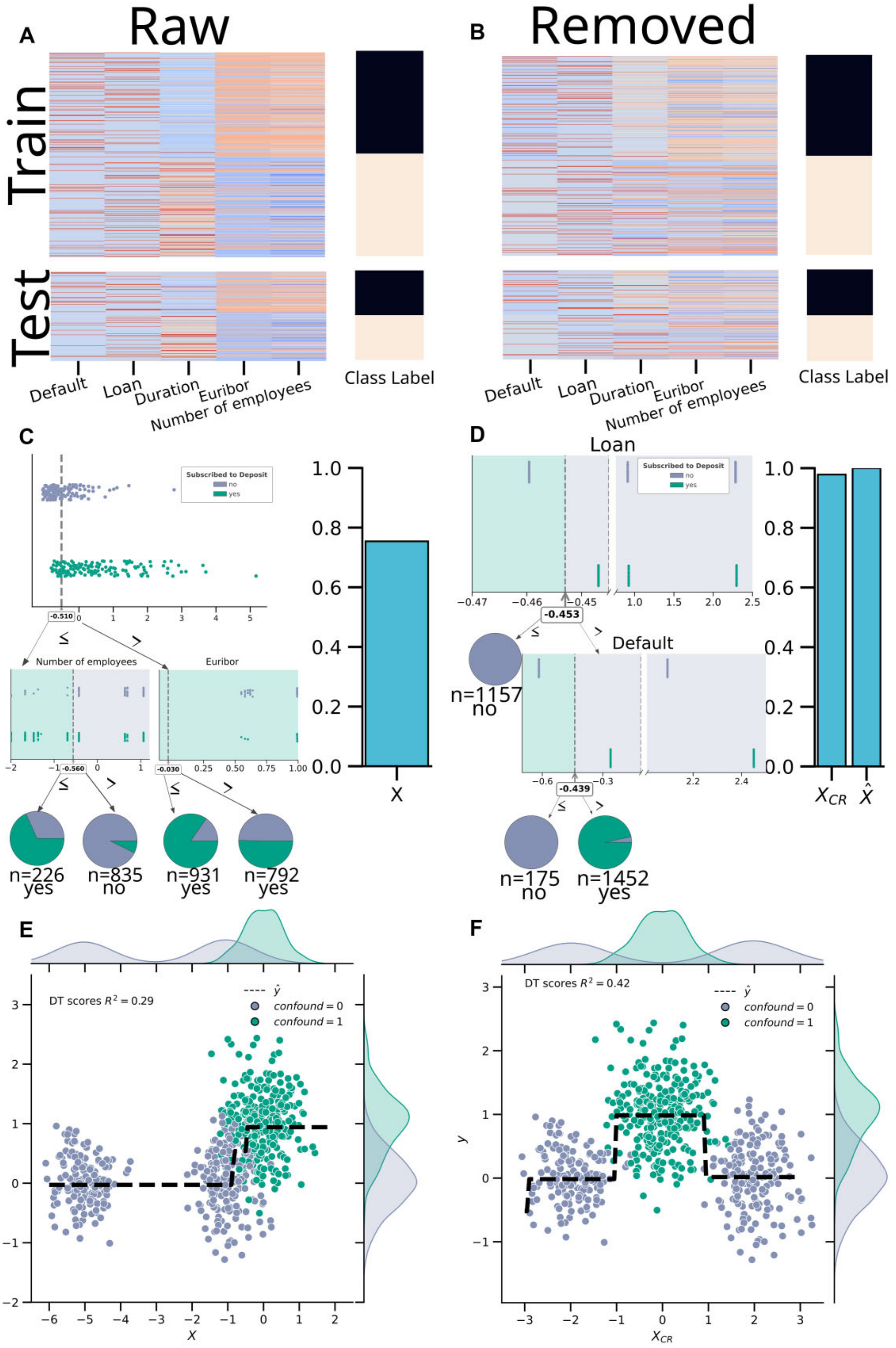

Figure 1: A walk-through analysis demonstrating our analysis pipeline and confound-leakage using DT. The results shown here are on the 30% test split. For the binary classification walk-through using the bank investment dataset, a subset of the features used is shown before CR (A) and after CR (B). Induced DTs and their performance before (C) or after CR (D). The DT after CR (D) is based on minute differences in only 2 features and still performs nearly perfectly and better compared to the DT on raw data (C). The regression analysis walk-through using simulated data is depicted as feature–target relationships with the dotted line showing the predicted values (E, F). The nonnormal distribution of the feature conditioned on the confound leaks information usable by the DT. Here, CR removes the linear relationship, as intended, but introduces a stronger nonlinear one by shifting the distribution of X_{CR} given *confound* = 0 in between the 2 peaks of X_{CR} given confound = 1 (F).

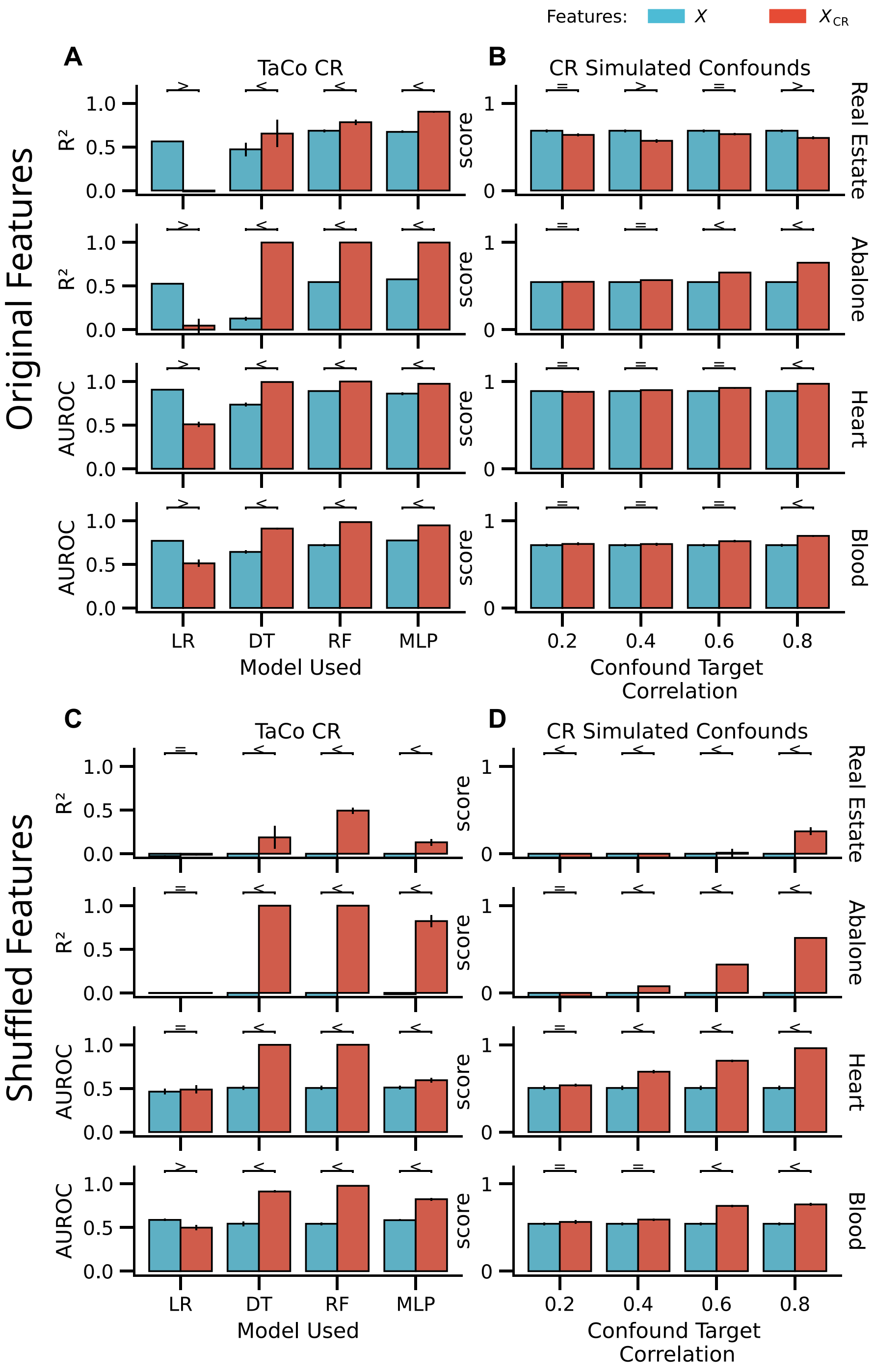

Figure 2: Performance on the UCI benchmark datasets when using raw vs. CR features (A) and raw vs. the predicted features given the confound/TaCo/$\hat{X}$ (B). The 2 columns correspond to (i) TaCo removal with 4 ML algorithms (logistic regression [LR], DT, RF, MLP) and (ii) CR with simulated confound with different correlations to the target (range 0.2–0.8) with RF. (A, B) Performance using the original features. (C, D) Performance on shuffled features. To check whether a difference between the performance of 2 models is meaningful, we used the Bayesian ROPE approach to identify what is most probable: performance being higher before removal (<), being higher after removal (>), or equivalent (=) (see the Methods section for details). When using a linear model (LR), TaCo removal leads to reduction in prediction performance, as expected. In contrast, nonlinear models lead to a higher performance for all datasets. This increase could be explained by confound removal revealing information already in the data (suppression) or confound removal leaking information into the features (confound-leakage). Shuffling the features destroys the association between features and the target; therefore, subsequent performance increase after TaCo removal indicates the possibility of confound-leakage (C, D). The simulated confounds show that an increase after CR is also possible for confounds weakly related to the target (B, D), and 1 dataset (Blood) shows strong evidence of confound-leakage.

(small) differences of the conditional distributions of features given the confound inside of CV-folds.

As DT-based models are very popular ML algorithms [28] and seem to be most susceptible to the described problems (see "TaCo removal increases performance of nonlinear methods"), we will focus on them in our simulations to decrease the complexity of our results. Furthermore, we will use a DT whenever there is only 1 features and RF when there are multiple features.

Confound-leakage due to deviation from normal distributions

Consider simulating a standard normal feature not informative of a binary target. Then consider adding a smaller distribution around opposing extreme values separately for each class of a binary target (Fig. 3A). The resulting feature only differs systematically w.r.t. the classes at the extreme values. As CR with a binary confound is equivalent to subtracting the mean for each confounding group from the respective feature, this operation is now biased toward the extreme parts of the feature distribution. Consequently, X_{CR} exposes confounding information in terms of decrease in the overlap of the feature distributions conditioned on the confound (Fig. 3A,B). In other words, confounding information leaked via CR in turn increases the prediction performance (AUROC from 0.51 before to 0.58 after TaCo removal). To show that the increased performance is not only due to better prediction of extreme values, we also tested the same model on a test set without the extreme values. The results were in line with previous observations, as the AUROC improved from 0.48 before to 0.57 after CR.

We also observed higher performance after similar decreased overlap due to TaCo removal in a simplified version of the "house pricing" UCI benchmark dataset (Fig. 3C,D), providing real-world evidence for this phenomenon.

Lastly, we investigated whether such effects could also occur when randomly sampling nonnormal distributed features instead of carefully constructing the features conditioned on the confound. To this end, we sampled an increasing number of features (1 to 100) either using a random normal or skewed (χ^2, $df = 3$) distribution independent of a normally distributed target.

Using RF, we observed increased performance after TaCo removal with skewed features but not with normally distributed features (e.g., R^2 of $M = 0.23$ with $SD = 0.06$ compared to R^2 of $M = -0.04$ with $SD = 0.04$, respectively, with 100 features). Importantly, this effect increased with the number of features (Fig. 4). To further illustrate this point, we performed another simulation depicting a typical confounding situation. Here, we sampled an increasing number of features (1 to 100) with different χ^2 distribution given a binary confound (df = 3 (4) and scale = 0.5 (1) for confound = 0 (1)). The target was sampled from a normal distribution ($M = 0$, $SD = 0.2$), and the confound was added to it. Analysis of these data shows an increased performance after confound removal from $M = -0.52$ ($SD = 0.02$) to $M = -0.50$ ($SD = 0.03$) using 1 feature and from $M = -0.02$ ($SD = 0.01$) to $M = 0.18$ ($SD = 0.01$) using 100 features. These results demonstrate that the effect of confound-leakage increases with increasing number of features. These simulations show that skewed features and, by extension, potentially other nonnormal distributed features can lead to confound-leakage. Interestingly, another consequence of nonnormal distributions is insufficient removal of confounding information [21].

Confound-leakage due to limited precision features

A similar effect was observed with binary features, where unbalanced feature distributions conditioned on the confound led to leakage. Using simulations, first we confirmed that a binary feature perfectly balanced in respect to the TaCo did not lead to confound-leakage (AUCROC of $M = 0.50$, $SD = 0$). Then, we repeated similar simulations but now we swapped 2 randomly selected distinct values of the feature within each CV-fold, preserving the marginal distribution of the feature but slightly changing its distribution conditional on the confound. This can be seen as adding a small amount of noise to the feature. Still, such a simple manipulation led to drastic leakage after TaCo removal with perfect AUCROC ($M = 1.00$, $SD = 0.00$), compared to AUCROC without CR ($M = 0.52$, $SD = 0$).

To further demonstrate this effect, we analyzed a simple demonstrative classification task using DT and 2 binary features derived from the UCI "heart dataset" representing the resting electrocardiographic (Restecg) results. Without CR, the DT had 117 nodes and achieved a moderate AUCROC ($M = 0.74$, $SD = 0.06$). In stark contrast, after TaCo removal, the DT was extremely simple with only 5 nodes and achieved near-perfect AUROC ($M = 0.99$, $SD = 0.01$) (Fig. 3E). Tellingly, this DT was able to make accurate predictions based on numerically minute differences in feature values. The reason for this becomes apparent when remembering that CR with a binary confound is equivalent to subtracting the mean of the corresponding confounding group from the respective feature. When applied to a binary feature, this results in 4 distinct values for a residual feature (Fig 3E). When taken together with the results on the benchmark UCI data (see "Analyses of benchmark data"), we can see that such minute differences can be exploited by models such as DTs, RFs, and MLPs but likely not by linear models. It is important to note that leakage through minute differences was observed for not only binary features but also other features with a limited precision (values containing only integers or with limited fractional parts). To demonstrate this, we predicted a random continuous target using either a normally distributed feature or the same feature rounded to the first digit. The original nonrounded feature performed at chance level both before (R^2: $M = -1.10$, $SD = 0.06$) and after TaCo removal (R^2: $M = -1.03$, $SD = 0.07$), while after rounding, it led to an improvement from $M = -0.08$ ($SD = 0.01$) to $M = 0.70$ ($SD = 0.16$) after TaCo removal. Features with limited precision (i.e., with no or rounded fractional part) are common, for instance, age in years, questionnaires in psychology and social sciences, and transcriptomic data.

Confound-leakage poses danger in clinical applications

ADHD is a common psychiatric disorder that is currently diagnosed based on symptomatology, but objective computerized diagnosis is desirable [29]. Ideally, a predictive model for diagnosing ADHD should not be biased by comorbid conditions (e.g., depression) [30]. To this end, comorbidity can be treated as a confound. However, a confound-leakage affected model, albeit with appealing performance, could lead to misleading diagnosis and treatment. To highlight the danger of confound-leakage on this clinically relevant task, we analyzed a dataset with speech-derived features with the task to distinguish individuals with ADHD from controls. Our version of the dataset is a balanced subsample of the dataset described by von Polier et al. [3].

The baseline RF model without CR provided mean AUROC ($M = 0.71$, $SD = 0.02$). We then removed 4 confounds commonly considered for this task—age, sex, education level, and depression score (Beck's Depression Inventory, BDI)—via featurewise CR in a CV-consistent manner. This resulted in a much higher AUCROC ($M =$

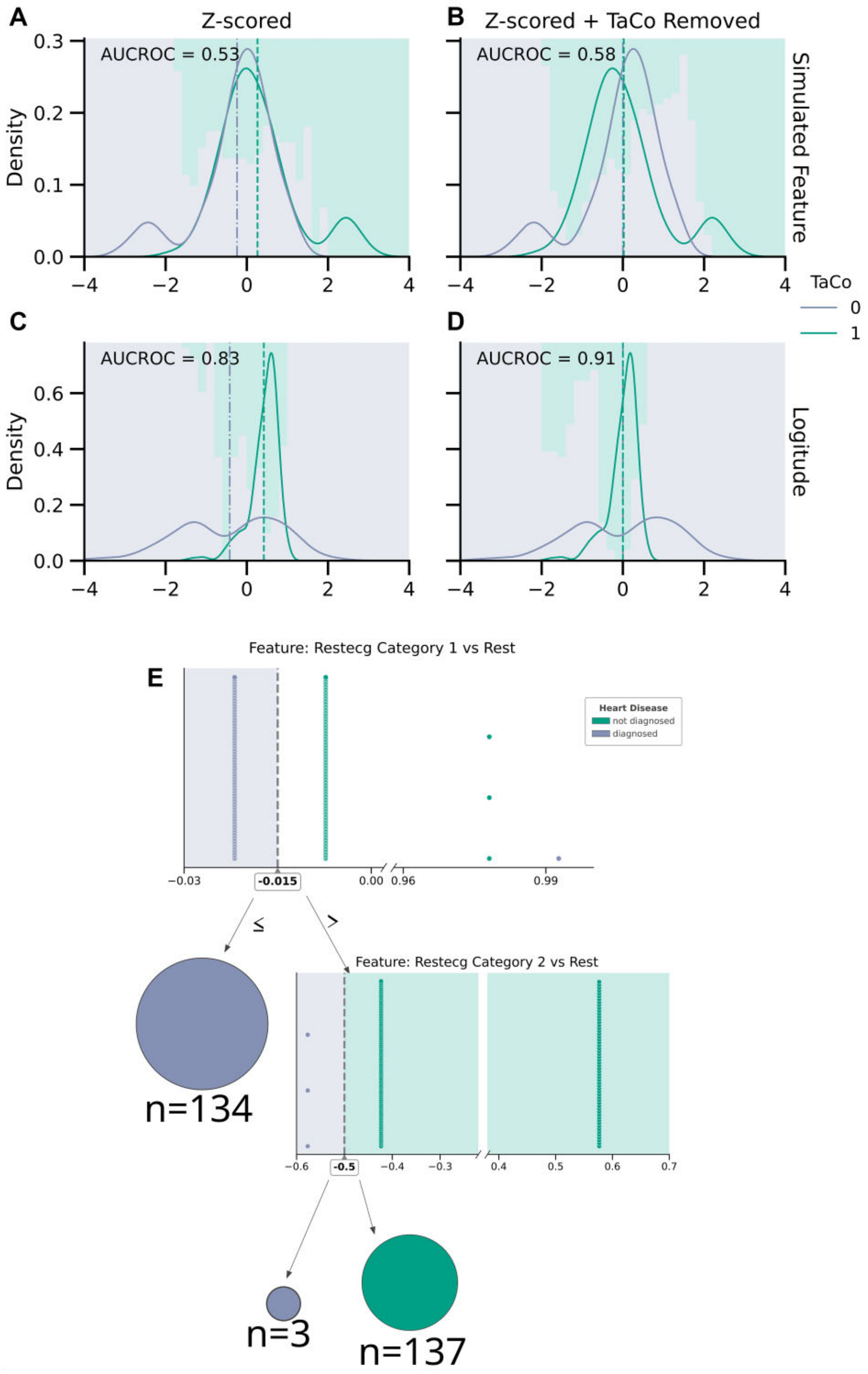

Figure 3: Two mechanisms for confound-leakage. First mechanism where nonnormal distributions get shifted apart through CR. (A, B) An example using a simulation with extreme values on opposing sides for 1 feature conditioned on the TaCo. (C, D) A simplified version (binary target for visualization purposes) of the house price UCI benchmark dataset. Here, the distributions of the feature conditional on the TaCo are different (C): a narrow distribution (TaCo = 1) and a distribution with 2 peaks (TaCo = 0). TaCo removal shifts the narrow distribution in between the two peaks (D), leaking information usable by nonlinear ML algorithms. The second mechanism, leakage through minute differences in the feature after CR, is highlighted through the visualization of the DT trained on the heart dataset after CR (E). Distribution plots visualize the data at each decision node. The decision boundary is shown as a dotted line. For decision nodes before leaf nodes, the side of the decision node leading into a prediction is colored to represent the predicted label as diagnosed (green) or not (purple). The minute differences in the 2 used features that perfectly separate the data into the 2 classes can be seen.

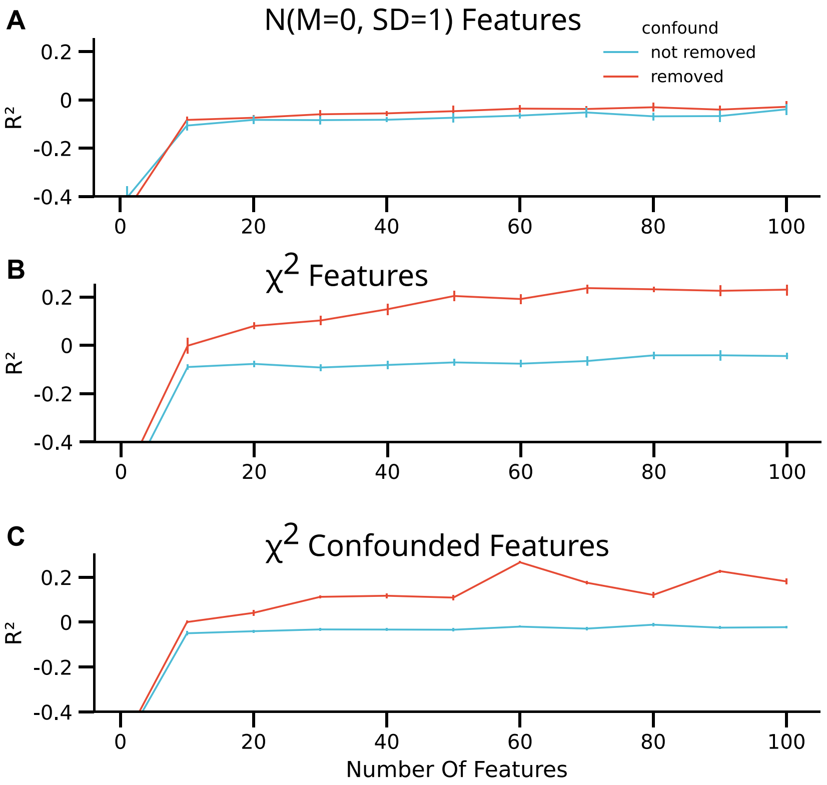

Figure 4: Prediction performance of an RF trained with (blue) or without (red) confound removal on an increasing number of features. Each feature was sampled from a random standard normal distribution ($M = 0$, $SD = 1$), a random χ^2 distribution with $df = 3$, or a χ^2 distribution with a $df = 3$, scale = 0.5 or $df = 4$, scale = 1 for the confound being equal to 0 and 1, respectively. (A) The RF trained on the normally distributed features did not achieve performance above the chance level ($R^2 < 0$) irrespective of confound removal. (B, C) When training the RF on either of the χ^2 distributed features, confound removal resulted in above chance-level performance ($R^2 > 0$). This effect increased with an increasing number of features and can only be explained by confound removal leaking information into the features.

0.86, $SD = 0.02$). This model would be very attractive for real-world application if its performance is true (i.e., not impacted by leakage). However, as we have shown with our analyses, confound-leakage can lead to such performance improvement. If confound-leakage is indeed driving the performance, then this model could misclassify individuals as having ADHD because of confounding effects (e.g., their sex or depression), leading to misdiagnosis and wrong therapeutic interventions. To disentangle the effect of each confound, we looked at the performance after CR for each confound separately. Performing CR with BDI led to a high AUCROC with original features after CR ($M = 0.91$, $SD = 0.01$) and shuffled features ($M = 0.84$, $SD = 0.01$) (Fig. 5A,B). This result revealed that BDI is driving the potential leakage, owing to its strong relation to the target (point-biserial correlation, $r = 0.61$, $P < 0.01$). Furthermore, a permutation test also led to the same conclusion (see Methods and Supplementary Fig. S2). Training CR models only on healthy individuals can be helpful in clinical applications [4]. We investigated this variant of CR, and again the AUCROC increased for original features after CR $M = 0.83$ ($SD = 0.02$) and an increase with shuffled features from $M = 0.51$ ($SD = 0.05$) to $M = 0.79$ ($SD = 0.02$), suggesting that confound leakage is also a concern for variants of CR. Lastly, we wanted to evaluate why we observe confound-leakage on this dataset. The limited precision of features cannot be the reason here as all features are continuous. Therefore, we hypothesized that the confound leaked due to some features deviating from normal distributions. To this end, we first compared the feature importance between the RF after CR and using the original features. Here, we observed the RFs' 10 most important features were completely different (Fig. 5C,D), indicating that the 2 RF models rely on different relationships in the data. Next we visualized the distributions of the 2 most important features of the RF after CR for both models. This visualization (Fig. 5E,F) clearly shows that CR has shifted the distributions due to deviations from normal distributions leaking information in their joint distribution. Furthermore, we trained new DTs using only these 2 features before or after CR. This led to an increase of AUCROC from 0.61 to 0.70 after CR only using these features. These analyses clearly demonstrate that real-world applications could suffer from confound-leakage and users should exercise care when implementing and validating a CR–ML workflow.

Discussion

Here, we exposed a hitherto unexplained pitfall in CR–ML workflows that use featurewise linear confound removal—a method popular in epidemiological and clinical applications. Specifically, we have shown this method can counterintuitively introduce confounding, which can be exploited by some nonlinear ML algorithms. Thus, in addition to the already known pitfalls of residual confounding [21], our results show that CR may actually introduce confounding information. We provide evidence of confound-leakage using a range of systematic controlled experiments on real and simulated data comprising both classification and regression tasks. First, to establish confound-leakage as opposed to information-reveal (of possibly nonlinear information) as the

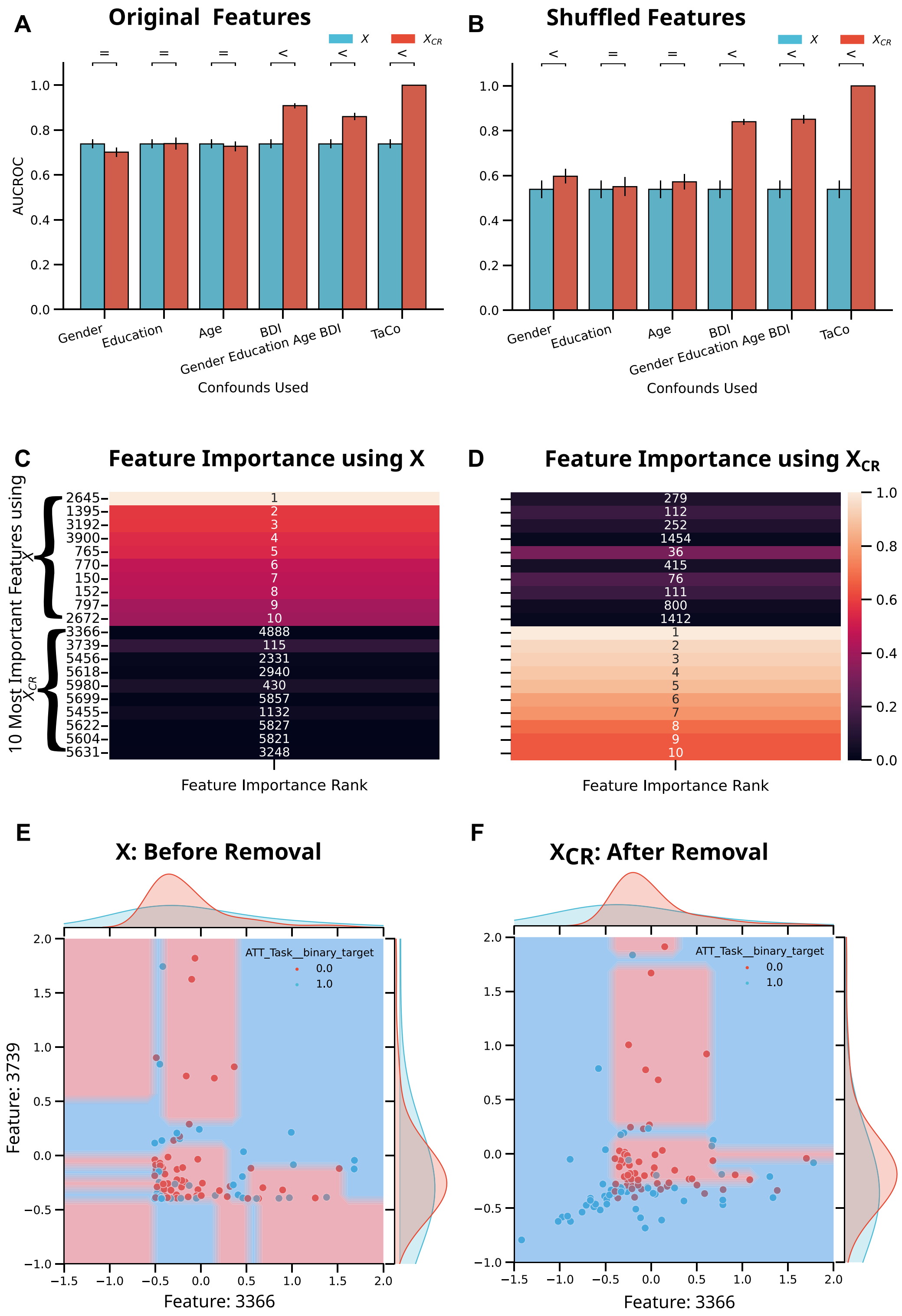

Figure 5: The real-world ADHD speech dataset. The performance when using different confounds (A, B), most important features of RF when using Beck's Depression Inventory (BDI) as confound (C, D), and visualization of confound-leakage due to deviation from normal distributions (E, F). (A) The performance of an RF predicting ADHD vs. healthy controls using the original features. To check whether a difference is meaningful, we used the Bayesian ROPE approach to identify what is most probable: performance being higher before removal (<), being higher after removal (>), or equivalent (=) (see Methods section). An increased performance can be observed when using all confounds, BDI as a confound, or the TaCo. The same pattern appears when the features were shuffled (B). This shows that the increase in performance is due to confound-leakage, and BDI is a driving factor for this leakage as it leaks information when used as a confound. (C, D) The 10 most important features for using X and X_{CR} as features. The feature ranking is shown as a white label on top of each cell. The most important features are different for X and X_{CR}. Furthermore, the most important features of 1 model ranked as very unimportant in the other. (E, F) Decision boundaries of DT trained on the 2 most important features after CR. The background colors indicate the prediction of the model, and the points show the true target value and the x-axis the 2 most important features. The distribution of each feature conditioned on the target is shown as the density plots. One can see that CR leaks information by cleanly separating the blue and red points.

reason behind increased performance after CR, we proposed the TaCo framework (i.e., using the target as "superconfound"). This extreme case of confounding allowed us to establish the existence, the extent, and possible mechanisms of confound-leakage. Specifically, by comparing the without CR baseline performance with CR after feature shuffling ($\tilde{X}_{CR}$), this framework can identify confound-leakage as the cause of increased predictive performance. We then extended the same framework to the more realistic scenario of weaker confounds showing that also there confound-leakage can occur.

To identify risk factors of confound-leakage, we performed several analyses. First, we demonstrated a mechanism by which confound-leakage can occur: differences of the conditional distributions of features given the confound. In the case of continuous features, nonnormal distributions (e.g., skewed distributions) and in the case of discrete features, frequency imbalances can cause leakage, although other mechanisms could exist. Additionally, we show that features of limited precision (e.g., age in years and counts) also showed susceptibility due to this mechanism. Lastly, our results showed that the risk of confound-leakage increases with the number of features, which is especially problematic in the era of "big data," where tens of thousands of features are a norm.

Still, we would like to highlight that we do not claim to have found all possible ways confound-leakage can happen. For instance, it is possible that other modeling approaches, even linear ones, could be susceptible to confound-leakage, although we did not find evidence for it in our analyses. Nonetheless, confound-leakage can bias the data and may negatively impact subsequent statistical analysis [21].

It is important to note that although similar, confound-leakage is not equal to collider bias. Colliders are variables causally influenced by both the features and target [19]. Both collider bias and confound-leakage describe situations where variable adjustment can lead to spurious relationships between features and target. However, the collider bias assumes that the removed variable has to be caused by both the features and the target, which is not shared by confound-leakage. One cannot exclude the possibility of collider removal using CR for many of our experiments as our operational definition of confounds does not include any assumption of causality. Still, we observe confound-leakage through CR for at least 1 causally defined confound (see "Walk-through analysis") and variables showing relationship only with the target. Such associations are not covered by the causal relationships described by a collider. In other words, the mechanisms of confound-leakage can lead to leaked information due to any variable related to the target and not only colliders or causal confounds.

Taken together, our extensive results show that the commonly used data types and settings of nonlinear ML pipelines are susceptible to confound-leakage when using featurewise linear CR. Therefore, this method should be applied with care, and the ensuing models should be closely inspected, especially in critical decision domains. We concretely demonstrated this using an application scenario from precision medicine by building models for diagnosis of ADHD. We found that the attempt to control for comorbidity with depression using CR led to confound-leakage. As many disorders often exhibit severe comorbidity (e.g., AHDH and depression, as we demonstrated here, but also neurodegenerative disorders are strongly confounded by aging-related factors [31] as well as comorbidity in mental disorders [32, 33]), the issue of confound-leakage should be carefully assessed in all such applications. We recommend the following best practices when applying CR together with nonlinear ML algorithms:

1) Assess confounding strength: Check the confounds' relation to each feature and the target. In general, confounds strongly related to the target pose a greater danger of leaking predictive information. Here, we used a straightforward approach of measuring the correlations between the confound and target/feature. Other methods can be employed (e.g., proposed by Spisak [27]). Furthermore, measuring how dependent the predictions of a model are on the confound by permutation testing [34, 35] or the approach proposed by Dinga et al. [21] can be helpful. To gain additional information, the reader might be interested in methods to estimate the variance in the target explained by ML predictions that confounds cannot explain [21, 27].
2) Compare performance with and without CR: If the performance increases after CR, one should investigate the reason behind the increase.
3) Gain evidence against or in favor of the confound-leakage: The procedure of shuffling the features followed by CR as we defined in the TaCo framework can provide clues regarding confound-leakage. Our shuffling approach can be seen as a single iteration of permutation testing. As our experiments suggest this is sufficient to obtain an indication of confound-leakage. However, a permutation test-based null distribution can quantify the variability and provide additional information. It is important to note, however, that while this can provide evidence for confound-leakage, we are not aware of a procedure to definitively exclude confound-leakage as an explanation.
4) Carefully choose alternatives: If confound-leakage seems probable, then consider alternative confound adjustment methods. Stratification [20, 36] is commonly in conventional ML or unlearning of confounding effects [37], which is common in deep learning and further general approaches that promote fairness [12, 38]. Note, however, that these procedures may also entail pitfalls. Hence, we caution researchers to exercise care when applying any confound adjustment protocol and to carefully consider limitations of the modeling approach used.

Conclusions and future directions

Important societal questions involving health and economic policy can be informed by applying powerful nonlinear ML models to large datasets. To draw appropriate conclusions, confounds must be removed without introducing new issues that cloud the results. In the present study, we performed extensive numerical experiments to gather evidence for confound-leakage. Using feature shuffling and predictions due to confound predicted features as proposed here, investigators can get an initial indication of whether their pipeline and data are susceptible to confound-leakage. We highlighted the conditions most likely to lead to leakage. Although we made progress on understanding these issues, there is no full-proof method for detecting and eliminating leakage. We hope our results prompt others to push further, perhaps expanding on the standard definition we adopted for confounds by introducing causal analyses. We hope our and allied efforts inform both researchers and practitioners who incorporate ML models into their data analyses. As a starting point, we suggest following the guidelines we provide to mitigate against confound-leakage.

Methods

Data

We analyzed several ML benchmark datasets from diverse domains to draw generalizable conclusions. To ensure reproducibility, most datasets come from the openly accessible UCI repository [39]. We included 5 classification tasks and 5 regression tasks with different sample sizes and numbers of features. All classification problems were binary or were binarized, and class labels were balanced to exclude biases due to class imbalance [40].

We also used one clinical dataset, a balanced subsample of the ADHD speech dataset described by von Polier et al. [3]. This data includes 126 individuals with 6,016 speech-related features, a binary target describing ADHD status (ADHD or control) and contains 4 confounds: gender, education level, age, and depression score measured using the BDI. For more information on the datasets, see Supplementary Table S1.

Confound removal

Confound removal was performed following the standard way of using linear regression models. Following the common practice, we applied CR to all the features. Specifically, for each feature, a linear regression model was fit with the feature as the dependent variable and the confounds as independent variables. The residuals of these models, that is, original feature minus the fitted values were used as confound-free features ($X_{CR} = X - \hat{X}$). This procedure was performed in a CV-consistent fashion (i.e., the confound removal models were fitted on the training folds and applied to the training and test folds) [20, 22].

Target as a confound (TaCo)

The TaCo framework allows systematic analysis of confound removal effects. Confounding is a 3-way relationship between features, confounds, and the target. This means that a confound needs to share variance with both the feature and the target. Measuring or simulating such relationships can be hard, especially if linear univariate relationships cannot be assumed. Furthermore, effects of confound removal should increase with the actual strength of the confound. The target itself explains all the shared variance and thus is the strongest possible confound. Therefore, using the target as a confound (i.e., TaCo) measures the most possible extent of confounding. In addition, using the TaCo simplifies the analysis to a 2-way relationship. Lastly, the TaCo approach is applicable to any dataset and can help to measure the strongest possible extent of confound-leakage even without knowing the confounds.

Machine learning pipeline

To study the effect of CR on both linear and nonlinear ML algorithms, we employed a variety of algorithms: linear/LR, linear kernel SVM, RBF kernel SVM, DT, RF, and MLP with a single hidden layer (relu). Additionally, we used dummy models to evaluate chance-level performance.

In the preprocessing steps, we normalized the continuous features and continuous confounds to have a mean of zero and unit variance, again in a CV-consistent fashion. Any categorical features were one-hot encoded following standard practice.

Evaluation

We compared the performance of ML pipelines with and without CR. To this end, we computed the out-of-sample AUCROC for classification and predictive R^2 from scikit-learn [41] for regression problems in a 10 times repeated 5-fold nested CV. We employed the Bayesian ROPE approach [42] to determine whether the results for a given dataset and algorithm with and without CR were meaningfully higher, lower, or not meaningfully different.

The Bayesian ROPE for model comparison

In this study, we used the Bayesian ROPE [42] approach to qualify differences between K-fold cross-validation results coming from 2 models. This approach uses the Bayesian framework to compute probabilities of the metric falling into a defined region of practical equivalence or of 1 ML pipeline scoring higher than the other. This is achieved by defining a region of equivalence (here we used 0.05). Consequently, the Bayesian ROPE approach allows us to make probabilistic statements regarding whether and, if so, which of the ML pipelines score higher. We summarize these differences using the following symbols: = (highest probability of pipelines scoring practically equivalent), < (highest probability of right pipeline scoring higher), and > (highest probability of left pipeline scoring higher). Other possibilities, such as the significance test correcting for the dependency structure in K-fold CV [43] or permutation testing by shuffling the target or features, can be employed when suitable.

Feature shuffling approach

Shuffling the features while keeping the confounds and target intact destroys the feature–target and feature–confound relationships while preserving the confound–target relationship. Therefore, after feature shuffling, any confound adjustment method cannot reveal the feature–target relationship, but it can still leak information. In other words, any performance above the chance level after CR on shuffled features is an indication of confound-leakage. Feature shuffling is also used in other approaches such as permutation testing (see section “The Bayesian ROPE for model comparison”) to test effectiveness of confound adjustment methods [21]. Permutation testing can be computationally expensive and, like other frequentist tests, it cannot accept the null hypothesis to establish equivalence. We, therefore, adopted a computationally feasible methodology. We shuffle the features, perform repeated nested cross-validation, and then apply the Bayesian ROPE. For completeness, we show that both permutation testing and the Bayesian ROPE detect confound leakage in the clinical dataset. In some cases, feature shuffling approaches might need further consideration, for instance, shuffling features within confound categories to preserve their joint distribution (see “Walk-through analysis”) and the possibility of suppression and leakage happening simultaneously. Nevertheless, they serve as a useful tool for detecting confound leakage, as shown in this work.

Availability of Source Code and Requirements

- Project name: Confound-leakage
- Project homepage: https://github.com/juaml/ConfoundLeakage
- Operating system(s): GNU/Linux
- Programming language Python 3.10.8 [43]
- Other requirements: scikit-learn 0.24.2, baycomp 1.0.2, matplotlib 3.5.1, seaborn 0.11.2, dtreeviz 1.3.5, numpy 1.22.3, pandas 1.2.5
- License: GNU Affero General Public License v3.0

Data Availability

All 10 UCI benchmark datasets can be accessed freely at the UCI machine learning repository [39]. Together with our simulated data (available under [44]), the UCI benchmark datasets compose minimal data sets to reproduce our key findings. Additionally, we analyzed 1 real-world clinical dataset [3]. These sensitive data are available from PeakProfiling GmbH with certain restrictions. Restrictions apply to the availability of the data, which were used under license for this study. Please contact Jörg Langner, the cofounder and CTO of PeakProfiling GmbH, with requests. An archival copy of the code and supporting data is also available via the *GigaScience* database, GigaDB [45].

Additional Files

Supplementary Fig. S1. Performance on the UCI benchmark datasets when using raw vs. CR features (A) and raw vs. the predicted features given the confound/TaCo/$\bar{X}$ (B). The 2 columns correspond to (i) TaCo removal with 6 ML algorithms (LR, DT, RF, MLP, Lin SVM, RBF SVM) and (ii) CR with simulated confound with different correlation to the target (range 0.2–0.8) with RF. (A, B) Performance using the original features. (C, D) Performance on shuffled features. When using a linear model (LR), TaCo removal leads to reduction in prediction performance, as expected. In contrast, nonlinear models lead to a higher performance for all datasets. This increase could be explained by confound removal revealing information already in the data (suppression) or confound removal leaking information into the features (confound-leakage). Shuffling the features destroys the association between features and the target; therefore, subsequent performance increase after TaCo removal indicates the possibility of confound-leakage (C, D). The simulated confounds show that an increase after CR is also possible for confounds weakly related to the target (B, D), and 1 dataset (Blood) shows strong evidence of confound-leakage.

Supplementary Fig. S2. We performed permutation testing with 1,000 iterations. After shuffling the features, a significantly lower performance was observed compared to the original features X. No significant difference between raw and shuffled features was observed when using the X_{CR} features. This result is in line with the leakage hypothesis as the higher accuracy after shuffling and CR indicates leaking target-related confounding information into the features.

Supplementary Table S1. Overview of all the datasets used. Shows each dataset with their associated problem type, sample size, feature number, and source. Our datasets cover a big range of features and sample sizes. All datasets with the exception of the speech ADHD one are freely accessible through the UCI machine learning repository.

Supplementary Table S2. Overview of all the simulations used. Including pseudo-code to create the features (X), target (y), and confounds (c). Variables were sampled from normal distributions (*N*), with different means (*M*) and standard deviations (*SD*) or binary distributions (binary). *repeat(list, number)* indicates the repetition of a list of values ([value, value,...]) are repeated for a number of times. concat means the concatenation of multiple arrays, and *where condition* → operation means that the operation is executed for where the condition is met.

Abbreviations

ADHD: attention-deficit/hyperactivity disorder; AUCROC: receiver operating characteristic curve; BDI: Beck's Depression Inventory; CR: confound regression; CV: cross-validation; DT: decision tree; LR: logistic regression; ML: machine learning; MLP: multilayer perceptron; RBF: radial basis function; RF: random forest; SVM: support vector machine; TaCo: target as a confound; UCI: UC Irvine.

Ethical Approval

All procedures contributing to this work comply with the ethical standards of the relevant national and institutional committees on human experimentation and with the Declaration of Helsinki of 1975, as revised in 2008. The ADHD data collection and use involving human subjects/patients were approved by the ethics committee of the Charite Universitatsmedizin Berlin, Berlin, Germany; the approval number is EA4/014/10. All necessary patient/participant consent has been obtained and the appropriate institutional forms have been archived by the data collectors. The ethics protocols for analyses of these data were approved by the Heinrich Heine University Düsseldorf ethics committee (No. 4039, 4096).

Competing Interests

The authors declare no competing financial or nonfinancial interests but the following personal financial interest: Georg G. von Polier participated and received payments in the national advisory board ADHD of Takeda.

Funding

This work was partly supported by the Helmholtz-AI project DeGen (ZT-I-PF-5-078), the Helmholtz Portfolio Theme "Supercomputing and Modeling for the Human Brain," and Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project-ID 431549029–SFB 1451 project B05.

Authors' Contributions

Study concept and design: S.H., B.C.L, G.G.P., S.W., H.S., S.B.E., and K.R.P. Data collection and processing: G.G.P. for the ADHD data. Data analysis and interpretation: all authors. Drafting of the manuscript: S.H. Critical revision of the manuscript for important intellectual content and final approval: all authors. Supervision: B.C.L., S.B.E., and K.R.P.

Acknowledgments

We thank the UCI machine learning repository [39] and the original dataset contributors.

References

1. Zeng LL, Wang H, Hu P, et al. Multi-Site diagnostic classification of schizophrenia using discriminant deep learning with functional connectivity MRI. EBioMedicine 2018;30:74–85. https://doi.org/10.1016/j.ebiom.2018.03.017.
2. Qin K, Lei D, Pinaya WHL, et al. Using graph convolutional network to characterize individuals with major depressive disorder across multiple imaging sites. eBioMedicine 2022;78:103977. https://doi.org/10.1016/j.ebiom.2022.103977.
3. von Polier GG, Ahlers E, Amunts J, et al. Predicting adult attention deficit hyperactivity disorder (ADHD) using vocal acoustic features. medRxiv. 2021. https://doi.org/10.1101/2021.03.18.21253108.

4. Dukart J, Schroeter ML, Mueller K. Age correction in dementia—matching to a healthy brain. PLoS One 2011;6:e22193. https://doi.org/10.1371/journal.pone.0022193.
5. Jo ES, Gebru T. Lessons from archives: strategies for collecting sociocultural data in machine learning. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. Association for Computing Machinery; New York, NY, USA. 2020: 306–16. https://doi.org/10.1145/3351095.3372829.
6. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics 2007;8:118–27. https://doi.org/10.1093/biostatistics/kxj037.
7. Whalen S, Schreiber J, Noble WS, et al. Navigating the pitfalls of applying machine learning in genomics. Nat Rev Genet 2022;23:169–81. https://doi.org/10.1038/s41576-021-00434-9.
8. Pomponio R, Erus G, Habes M, et al. Harmonization of large MRI datasets for the analysis of brain imaging patterns throughout the lifespan. Neuroimage 2020;208:116450. https://doi.org/10.1016/j.neuroimage.2019.116450.
9. Badgeley MA, Zech JR, Oakden-Rayner L, et al. Deep learning predicts hip fracture using confounding patient and healthcare variables. NPJ Digit Med 2019;2:31. https://doi.org/10.1038/s41746-019-0105-1.
10. Luders E, Toga AW, Thompson PM. Why size matters: differences in brain volume account for apparent sex differences in callosal anatomy: the sexual dimorphism of the corpus callosum. Neuroimage 2014;84:820–4. https://doi.org/10.1016/j.neuroimage.2013.09.040.
11. Wiersch L, Hamdan S, Hoffstaedter F, et al. Accurate sex prediction of cisgender and transgender individuals without brain size bias. Sci Rep 2023;24;13(1):1386824. https://doi.org/10.1038/s41598-023-37508-z.
12. Mehrabi N, Morstatter F, Saxena N, et al. A survey on bias and fairness in machine learning. ACM Comput Surv 2021;54:1–35. https://doi.org/10.1145/3457607.
13. MacKinnon DP, Krull JL, Lockwood CM. Equivalence of the mediation, confounding and suppression effect. Prev Sci 2000;1:173–181. https://doi.org/10.1023/A:1026595011371.
14. Pourhoseingholi MA, Baghestani AR, Vahedi M. How to control confounding effects by statistical analysis. Gastroenterol Hepatol Bed Bench 2012;5:79–83. http://doi.org/10.22037/ghfbb.v5i2.246.
15. Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009;248–55. https://doi.org/10.1109/CVPR.2009.5206848.
16. Alfaro-Almagro F, McCarthy P, Afyouni S, et al. Confound modelling in UK Biobank brain imaging. NeuroImage 2021;224:248–255. https://doi.org/10.1016/j.neuroimage.2020.117002.
17. Rao A, Monteiro JM, Mourao-Miranda J. Predictive modelling using neuroimaging data in the presence of confounds. NeuroImage 2017;150:23–49. https://doi.org/10.1016/j.neuroimage.2017.01.066.
18. Chyzhyk D, Varoquaux G, Milham M, et al. How to remove or control confounds in predictive models, with applications to brain biomarkers. GigaScience 2022;11:giac014. https://doi.org/10.1093/gigascience/giac014.
19. Greenland S. Quantifying biases in causal models: classical confounding vs collider-stratification bias. Epidemiology 2003;14:300–306. https://doi.org/10.1097/00001648-200305000-00009.
20. Snoek L, Miletić S, Scholte HS. How to control for confounds in decoding analyses of neuroimaging data. NeuroImage 2019;184:741–760. https://doi.org/10.1016/j.neuroimage.2018.09.074.
21. Dinga R, Schmaal L, Penninx BWJH, et al. Controlling for effects of confounding variables on machine learning predictions. bioRxiv. 2020. https://doi.org/10.1101/2020.08.17.255034.
22. More S, Eickhoff SB, Caspers J, et al. Confound removal and normalization in practice: a neuroimaging based sex prediction case study. Lecture Notes Comput Sci 2021;12461:3–18. https://doi.org/10.1007/978-3-030-67670-4_1.
23. Weele TJV, Shpitser I. On the definition of a confounder. Ann Stat 2013;41:196–220. https://doi.org/10.1214/12-AOS1058.
24. Dagaev N, Roads BD, Luo X, et al. A too-good-to-be-true prior to reduce shortcut reliance. Pattern Recog Lett 2023;166:164–71. https://doi.org/10.1016/j.patrec.2022.12.010.
25. Geirhos R, Jacobsen JH, Michaelis C, et al. Shortcut learning in deep neural networks. Nat Mach Intell 2020;2:665–73. https://doi.org/10.1038/s42256-020-00257-z.
26. Nygaard V, Rødland EA, Hovig E. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. Biostatistics 2016;17:29–39. https://doi.org/10.1093/biostatistics/kxv027.
27. Spisak T. Statistical quantification of confounding bias in machine learning models. Gigascience 2022;11:giac082. https://doi.org/10.1093/gigascience/giac082.
28. Grinsztajn L, Oyallon E, Varoquaux G. Why do tree-based models still outperform deep learning on typical tabular data? Adv Neural Inform Process Syst 2022;35:507–20. https://doi.org/10.48550/arXiv.2207.08815.
29. Gualtieri CT, Johnson LG. ADHD: is objective diagnosis possible? Psychiatry 2005;2:44–53. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2993524/.
30. Katzman MA, Bilkey TS, Chokka PR, et al. Adult ADHD and comorbid disorders: clinical implications of a dimensional approach. BMC Psychiatry 2017;17:302. https://doi.org/10.1186/s12888-017-1463-3.
31. Wyss-Coray T. Ageing, neurodegeneration and brain rejuvenation. Nature 2016;539:180–6. https://doi.org/10.1038/nature20411.
32. Joshi G, Wozniak J, Petty C, et al. Psychiatric comorbidity and functioning in a clinically referred population of adults with autism spectrum disorders: a comparative study. J Autism Dev Disord 2013;43:1314–25. https://doi.org/10.1007/s10803-012-1679-5.
33. Plana-Ripoll O, Pedersen CB, Holtz Y, et al. Exploring comorbidity within mental disorders among a danish national population. JAMA Psychiatry 2019;76:259–70. https://doi.org/10.1001/jamapsychiatry.2018.3658.
34. Epstein MP, Duncan R, Jiang Y, et al. A permutation procedure to correct for confounders in case-control studies, including tests of rare variation. Am J Hum Genet 2012;91:215–23. https://doi.org/10.1016/j.ajhg.2012.06.004.
35. Chaibub Neto E, Pratap A, Perumal TM, et al. A permutation approach to assess confounding in machine learning applications for digital health. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining KDD '19. New York, NY: Association for Computing Machinery; 2019:54–64. https://doi.org/10.1145/3292500.333090.
36. McNamee R. Regression modelling and other methods to control confounding. Occup Environ Med 2005;62:500–506. https://doi.org/10.1136/oem.2002.001115.
37. Dinsdale NK, Jenkinson M, Namburete AIL. Deep learning-based unlearning of dataset bias for MRI harmonisation and confound

removal. NeuroImage 2021;228:117689. https://doi.org/10.1016/j.neuroimage.2020.117689.

38. Zhao Q, Adeli E, Pohl KM. Training confounder-free deep learning models for medical applications. Nat Commun 2020;11:6010. https://doi.org/10.1038/s41467-020-19784-9.
39. Dua D, Graff C. UCI Machine Learning Repository. 2017. https://archive.ics.uci.edu/citation.
40. Collell G, Prelec D, Patil KR. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. Neurocomputing 2018;275:330–340. https://doi.org/10.1016/j.neucom.2017.08.035.
41. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. J Machine Learn Res 2011:12 (85): 2825–30. http://jmlr.org/papers/v12/pedregosa11a.html.
42. Benavoli A, Corani G, Demšar J, et al. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. J Machine Learn Res 2017;18:1–36. http://jmlr.org/papers/v18/16-305.html.
43. Guido Van R, Python Development Team. Python Tutorial: Release 3.6.4. 2018.1680921606. https://dl.acm.org/doi/10.5555/3217518.
44. Hamdan S, Love B, von Polier G, et al. Supporting code for “Confound-leakage: Confound Removal in Machine Learning Leads to Leakage.” *GitHub*. 2023. https://github.com/juaml/ConfoundLeakage.
45. Hamdan S, Love B, von Polier G, et al. Supporting data for “Confound-leakage: Confound Removal in Machine Learning Leads to Leakage.” GigaScience Database. http://dx.doi.org/10.5524/102420.

Received: January 6, 2023. **Revised:** June 1, 2023. **Accepted:** August 17, 2023