Mitigating Shortcut Learning Bias in ML: A Case Study in Energy Consumption Prediction

Matthew Caron
University of Paderborn, matthew.caron@uni-paderborn.de

Johannes Kriebel
University of Muenster, johannes.kriebel@wiwi.uni-muenster.de

Oliver Müller
University of Paderborn, oliver.mueller@uni-paderborn.de

The application of machine learning (ML) methods has become widespread in many disciplines. However, several methodological pitfalls in ML-based research have also become known in recent years. An issue that warrants our special attention is the phenomenon of shortcut learning. Shortcuts are decision rules that perform well on standard benchmark datasets but fail to generalize to more challenging situations. Consequently, ML models that rely heavily on shortcuts could fail when making inferences using data only slightly different from their original data. The phenomenon of shortcut learning represents a challenge to the reproducibility of ML-based studies and the generalizability of research results to real-world contexts. In this work, we examine a case of shortcut learning in energy consumption prediction theoretically and empirically. Using simulated and real data, we demonstrate that shortcut learning can lead to overestimation of performance metrics and model bias. We present a set of suggestions to identify and prevent shortcuts.

1. Introduction

Driven by recent algorithmic advances and the increasing availability of big data, the use of machine learning methods for predictive modeling is becoming more and more widespread across many scientific disciplines, such as information systems, management, marketing, finance, economics, and operations research (Shmueli & Koppius, 2011; Müller et al. 2017, Padmanabhan et al. 2022).

Yet, several researchers have started to voice concerns about biases, overoptimism, and reproducibility of ML-based research (e.g., Lin et al., 2013; Yang et al., 2018; Kapoor & Narayanan, 2022). For example, an open science initiative led by researchers from Princeton University found 20 reviews across 17 scientific fields that found errors in 329 papers that use ML-based science (Kapoor & Narayanan, 2022). The underlying reasons for these alarming observations are manifold (incl. a combination of lack of training, media hype about AI, and publication pressure), but one concrete source for pitfalls are the subtle differences between classical explanatory modeling (incl. statistical null hypothesis testing) and ML-based predictive modeling (esp. supervised ML) (Shmueli & Koppius, 2011). These differences make the performance evaluation of ML models tricky (Kapoor & Narayanan, 2022). An issue that warrants our special attention here is the recently discovered phenomenon of *shortcut learning*. In the context of ML, shortcuts are learned decision rules that perform well on standard benchmark datasets in the lab, but fail to generalize to more challenging real-world situations (Geirhos et al., 2020). ML models that rely heavily on shortcuts fail when making inferences using data only slightly different from their original training data, thereby posing a threat to the external validity of an ML-based study.

The issue of shortcut learning has been first observed in deep neural networks processing image or text data (Geirhos et al., 2020). For example, in a now-famous study, medical researchers reported that an ML classifier for detecting pneumonia from X-ray scans worked accurately in the hospital it was developed but poorly for scans from novel hospitals (Zech et al. 2018). The explanation was that the model had learned to identify particular X-ray systems by detecting a hospital-specific metal token on the scans. By combining the token with the hospital's uncommon pneumonia prevalence rate, the ML model achieved good prediction accuracy without learning anything about pneumonia at all. In other words, the ML model learned to take shortcuts.

Since its discovery, various methods for understanding, detecting, and avoiding shortcut learning have been proposed by the ML community (e.g., repositioning objects and using images with different backgrounds). However, they all focus on computer vision and natural language processing applications of deep neural networks.

In this paper, we (1) empirically show that shortcut learning is also a severe problem for ML on tabular numerical and categorical data, especially panel data, which is prevalent in business and economics research, and (2) propose and evaluate methods for detecting and mitigating the risk of shortcut learning.

Using simulated data and a well-known real-world dataset from energy informatics, we show that shortcut learning can lead to dramatic overestimation of the generalization capabilities of ML models and demonstrate ways to mitigate this risk.

Predicting energy consumption is a crucial and timely problem with significant implications. Accurate energy consumption prediction plays a pivotal role in ensuring optimal resource planning, modeling energy markets, and sustainable development. By forecasting energy demand patterns, policymakers, utility companies, and industries can make informed (investment) decisions, implement efficient energy distribution strategies, and proactively address energy challenges, ultimately contributing to reduced environmental impact and enhanced energy security for a sustainable future.

As the use of ML as a research method in business and economics is likely to increase in the future, we believe that raising awareness about shortcut learning as a potential threat to the external validity and reproducibility of ML-based studies and proposing guidelines and methods to mitigate this issue as good as possible can further help researchers understand the potential risks of using ML in their research and thus improve the validity and robustness of their findings.

The remainder of this paper is structured as follows: Section 2 presents some theoretical background on shortcut learning. Section 3 studies the prevalence of shortcut learning based on a simulation study and energy consumption data. Section 4 discusses the detection and mitigation of shortcuts and Section 5 concludes.

2. Theoretical Background

2.1. Defining Shortcut Learning

In order to better understand under which circumstances ML models rely on shortcuts, it is crucial to first understand the notion of generalization. In the context of ML, generalization can be defined as a system's aptitude to deal with scenarios that it has not previously encountered (Chollet 2019). There exist various degrees of generalization, ranging from local to extreme generalization (Chollet 2019). The form of generalization that most researchers and practitioners are primarily concerned with is known as *local* generalization, or robustness, referring to a system's ability, given that it has been trained on a sufficiently large dataset, to handle new samples that emerge from the same data-generating process – i.e., the new samples come from the same probability distribution as the training samples. *Broad* generalization, also referred to as flexibility, pertains to a system's capacity to handle unexpected situations that the system creators could not have anticipated. Hence, flexibility refers to the ability of an ML system to handle new data samples that come from a different distribution than the training samples. Arguably, many present-day ML systems fail at this task. Finally, *extreme* generalization pertains to a system's ability to handle entirely novel tasks that possess only abstract commonalities with previously encountered situations; or, in Chollet's words: "[the] adaptation to unknown unknowns across an unknown range of tasks and domains" (Chollet 2019, p.11). Currently, only humans are capable of such levels of generalization.

Geirhos et al. (2020) argue that shortcut learning is a key reason ML systems often fail to demonstrate broad generalization capabilities. Simply put, shortcuts are solutions that work well on

standard benchmark problems but fail to generalize when applied to more realistic situations, such as real-world scenarios (Geirhos et al. 2020, p.1). From a theoretical standpoint, ML algorithms learn a function, or a set of decision rules, that can map one or multiple input variables to one or multiple output variables. The learned decision rules can be arranged in a continuum ranging from (a) uninformative rules to (b) overfitting rules to (c) shortcut rules to (d) intended rules (Geirhos et al. 2020). Uninformative rules are decision rules that fail to achieve good performance on the data they were initially derived from and, as a result, represent ineffective solutions to a given problem. If a set of decision rules yields good performance on the data it was trained on but not on an independent and identically distributed (i.i.d.) test set, the ML system is said to suffer from overfitting (Ng 1997). A model that performs well on both the training and i.i.d. test sets, but fails on out-of-distribution (o.o.d.) samples, is said to apply shortcut rules. Such models typically score high on standard benchmark datasets in lab conditions or competitions but fail in real-world situations, where data distributions typically change or drift over time. Hence, the ML is not able to perform broad generalization. Finally, there are decision rules that work well on i.i.d. samples as well as on o.o.d. samples. These models are said to have learned the intended rules and, hence, also perform well in unexpected situations.

2.2. Sources of Shortcuts

Shortcut learning has mostly been identified for image data, and occasionally for text or audio data. For images, typical examples include ML models that learned to identify objects in images by identifying distinctive backgrounds (Zhu et al. 2017), positions (Rosenfeld et al. 2018), or textures (Geirhos et al. 2018) of objects, rather than the objects themselves. One can argue that these cases are instances of confounding, a much-discussed threat to the internal validity of causal models (Schölkopf 2022). While confounding represents a serious problem for deriving causal statements

from data-driven models, it is not necessarily a problem for predictive models, as long as the spurious correlations learned by the model are stable across populations and over time.

Another more subtle and lesser known source of shortcuts is clustering of observations and unobserved cluster characteristics. Here, ML models learn to identify groups of cases that have distinct characteristics with regards to the response variable of a model. The aforementioned pneumonia detection model is an example of this (i.e., the model learned to recognize scans from certain X-ray systems used in hospitals with particularly high pneumonia rates). Other examples include ML-based text classifiers that learned to classify texts by identifying biased annotators of texts instead of the annotations themselves (Geva et al. 2019). The groups of cases can also be of a temporal nature. For example, external shocks can lead to temporal shifts in the response variable of a model that cannot be anticipated from the information available at training time. For instance, in the above-mentioned hospital example, changes of local healthcare policies could lead to shifts in the distribution of patients between hospitals and, in turn, to the prevalence rates of certain diseases in certain hospitals. The performance of an ML model, which would use the identity of a hospital, department, or machine as a shortcut in order to detect diseases, would very likely drop as a consequence.

To summarize, when there are commonalities in groups or time periods which introduce dependencies between cases, it is possible that ML models do not learn to relate the features of the cases to the response variable, but rather learn to identify groups and time periods which might differ in quality. The functional relationships to distinguish these groups and time periods might then widely deviate from the actual relationships between the features and the response. Given that these groups and time periods might be easier to identify, the performance metrics calculated on i.i.d. test data could be heavily misleading. Models that learned to distinguish groups and time periods will also likely display a weak performance on new groups and time periods (o.o.d. test data).

2.3. Shortcut Learning in Tabular Data

To the best of our knowledge, the problem of shortcut learning has not been discussed in the context of tabular data. In the remainder of this research note, we assume a panel data structure such as the one visualized in Figure 1. The dataset contains observations of multiple entities (e.g., companies) over multiple time periods (e.g., years). We will show that this combination of entity groups and time periods makes tabular panel data prone to shortcut learning.

In general, in order to evaluate the predictive performance of an ML system, it is standard practice to train and assess its performance on several disjoint data samples (Friedman et al. 2009). The most fundamental sampling approach consists of randomly splitting a dataset into training and test sets. Usually, a random split leads to samples in the training and test sets that are sufficiently similar to each other or, in mathematical terms, that are drawn from the same probability distribution. Hence, the test set is said to be independent and identically distributed with regard to the training set (i.i.d.). However, as illustrated on the left of Figure 1, for complex multi-dimensional datasets random sampling might overlook possible temporal patterns that may be present in the data (e.g., economic expansion or contraction) or similarities within groups of entities (e.g., observations referring to the same company, industry, or geographical region).

For datasets with temporal dependencies, there is a growing awareness that the use of random sampling is impracticable due to the risk of leakage (Kaufman et al. 2012). Simply put, leakage refers to situations in which a model has been trained on information that would realistically not be available at the time of prediction – e.g., predicting a firm's credit rating in Q3 2022 based on a model trained on data from Q4 2022. Leakage leads to over-optimistic assessments of a models' predictive performance. To resolve this problem and avoid look-ahead bias, one typically splits the data in a time-based manner, as illustrated in the middle of Figure 1, to ensure that all training samples are anterior to the testing samples (Hyndman and Athanasopoulos 2018).

Detecting dependencies related to groups of entities is often much harder in complex multi-dimensional datasets. While in econometrics it is common to explicitly control for group dependencies through fixed effects (e.g., company or industry dummies), there seems to be less awareness about the risk of shortcut learning when group dependencies are present in ML. Therefore, sampling strategies that take groups of entities into consideration are less common. A notable exception are spatial cross-validation strategies in the geosciences, where models are trained on observations from one geographical region and tested on observations of a held-out region (Ploton et al., 2020; Beigaitė et al., 2022). Figure 1 (right) illustrates how this idea can be transferred to an entity-based grouping.

Figure 1 Random, Time-based, and Entity-based Sampling

In order to assess how commonly the above-discussed sampling strategies are considered, we conducted a literature review of papers published in journals from the AIS Senior Scholars' List of Premier Journals that applied predictive modeling in their research process.

Our review contained 228 articles published over the past 15 years. The key results are presented in Table 1. The table distinguishes between cross-sectional and panel data settings, as well as the above introduced sampling strategies. Most studies working with cross-sectional data apply random sampling. For panel data, time-based sampling is the most common sampling strategy; yet, about a fourth of these studies used random or entity-based sampling, which introduces the

risk of overlooking a shortcut learning bias. Over both types of data, entity-based sampling was least common and no study used a combined time- and entity-based sampling.

Table 1 Common Sampling Strategies in ML-Based Research

	Random	Time-based	Entity-based	Other	*Total*
Cross-Sectional data	148	23	8	6	*183*
Panel data	11	31	3	0	*45*
Total	*159*	*54*	*11*	*6*	*228*

3. Detecting Shortcut Learning Bias in Tabular Data

Shortcut learning often goes unnoticed, because it is difficult to detect with the testing procedures that are common today. Note that all of the above-introduced sampling strategies are only proxies for measuring an ML model's underlying ability to predict phenomena in *future* data. Yet, as all sampling strategies inevitably have to rely on historical data, it is difficult to judge whether a model relies on shortcuts or uses intended features and decision rules for making predictions. In the following sections, we will introduce a novel sampling strategy that has the potential to uncover shortcut learning and demonstrate it with simulated data.

3.1. Detection through Combined Time- and Entity-based Sampling

Our proposed sampling strategy to detect shortcut learning is inspired by the practice of comparing an ML model's predictive performance on training and test data in order to detect the well-known issue of overfitting. If a model's predictive accuracy is higher on training data than on independent and identically distributed test data, the model is said to overfit. A model that overfits relies on spurious associations in the training data that are not present in the test data. Recall that the definition of shortcut learning is that a model performs well on both the training and i.i.d. test sets, but fails on out-of-distribution samples. Hence, we propose to compare a model's predictive accuracy between an i.i.d. test set and an o.o.d. test set in order to detect shortcut learning. To

generate a test set that is sufficiently out-of-distribution, we combine the above-discussed, but rarely applied, time- and entity-based sampling strategies, leading to the combined sampling strategy depicted in Figure 2. In other words, we propose to train an ML model on just a sample of entities over a well-defined historical time period and test it on successive data of held-out entities. If the model's performance is substantially lower on the o.o.d. test set than on an i.i.d. test, generated for example through random sampling, it suffers from shortcut learning bias.

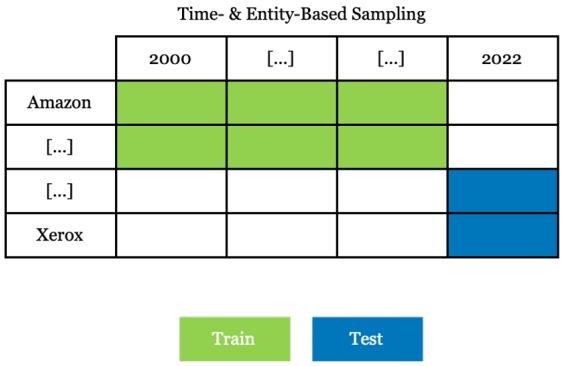

Figure 2 Combined Time- and Entity-based Sampling

3.2. Demonstration with Simulated Data

To validate that the proposed combined time- and entity-based sampling strategy indeed detects shortcut learning, we provide an illustrative numerical analysis based on simulated data and the evaluation on energy consumption data in the following.

3.2.1. Setup of Simulation. The simulation was conducted as follows. We generated three observable independent input variables, denoted as x_1, x_2, and x_3, and one observable dependent output variable, denoted as y. The data points are additionally associated with unobserved *entities* and *time* period variables. Each unique entity and time period has a distinct fixed influence on y, which is expressed by γ_e and γ_t. γ_e and γ_t are randomly drawn from a normal distribution

with mean zero and a standard deviation of 10 for each entity and time period, respectively. y is further related to x_1, but not to x_2 and x_3. Hence, we simulated the data in such a way that y is influenced solely by x_1, γ_e, and γ_t, with x_2 and x_3 having no effect on y. The value of y is further mostly determined by the entity and the time period, while x_1 has a limited effect on y. These relationships are formalized in the following equation, which formed the foundation for our data generating process:

$$y = x_1 + \gamma_e + \gamma_t \tag{1}$$

Further, the *entity* has *randomly* chosen effects on x_1 and x_2 that are a draw (η in x_1 and μ in x_2) from a normal distribution for each *entity*, while *time* has *randomly* chosen effects (τ) on x_3 that are a draw from a normal distribution for individual time periods. Apart from the effect of *entity* and *time*, the dispersion of x_1, x_2, and x_3 adheres to a standard normal distribution and is given by ϵ_1, ϵ_2, and ϵ_3.

$$x_1 = \eta + \epsilon_1 \tag{2}$$

$$x_2 = \mu + \epsilon_2 \tag{3}$$

$$x_3 = \tau + \epsilon_3 \tag{4}$$

Figure 3 visualizes the resulting simulated data. From the color coding, one can clearly see distinct clusters of data points which differ in their level of y. The level of y further differs between the time periods, as is visible in a comparison between the left and right panel of Figure 3. Note that there is no visible relationship between y and x_2, and even the existing relationship between y and x_1 is not clearly visible in the plots.

3.2.2. Training of ML Models. Eyeballing the plots without being aware of the existence of entities and time periods, one could think that the clusters with varying levels of y stem from

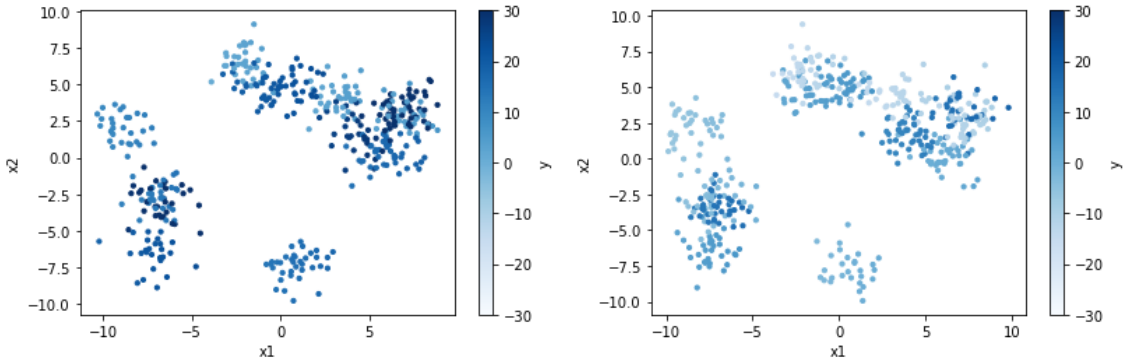

Figure 3 **Relation between x_1, x_2, and y for a time period with high y (left panel) and low y (right panel).**

complex non-linear relationships between y and x_1, x_2, and x_3 (the latter not being displayed in the plot). A flexible ML model that is able to consider interactions and non-linearities could be able to accommodate these complex relationships and therefore provide greater predictive accuracy than an additive linear model. However, due to the random variation in the association between entities and time periods and the dependent variable, any such identified relationships are likely shortcuts that work only on i.i.d. data, but not on o.o.d. data, e.g., *new* entities and time periods.

To empirically demonstrate this issue, we trained a Lasso (least absolute shrinkage and selection operator) and a Random Forest regression model both tasked with predicting the value of y based on the predictors x_1, x_2, and x_3 using the i.i.d and o.o.d. sampling strategies described earlier. Table 2 summaries the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) for both models over all four sampling strategies. Several interesting observations can be drawn from the performance data. First, the Random Forest model, which is able to capture interactions and non-linear relationships, dominates the additive and linear Lasso model for all sampling strategies. This is interesting, because the relationships between the considered predictors and the outcome were designed to either be perfectly linear (x_1) or non-existent (x_2 and x_3). Second, for both ML models the predictive performance on o.o.d. data is substantially

Table 2 Performance of different ML models trained with different sampling strategies on the simulated data

	Sampling Strategy	Evaluation		
		MAE	$RMSE$	R^2
Lasso	i.i.d. (test)	8.490	10.465	0.102
	o.o.d. (time)	9.065	11.049	0.084
	o.o.d. (entity)	14.081	16.826	-0.157
	o.o.d. (time/entity)	17.127	19.819	-0.116
Random Forest	i.i.d. (test)	6.552	8.572	0.397
	o.o.d. (time)	7.839	9.751	0.286
	o.o.d. (entity)	16.188	19.113	-0.494
	o.o.d. (time/entity)	19.358	22.462	-0.433

lower than on i.i.d. data. Third, the relative drop in performance is higher for the more flexible Random forest model compared to the simpler Lasso model. Taken together, these points support the suspicion that both models learned to take shortcuts by implicitly identifying the unobserved entity and time period clusters via their associations with x_1 and x_2 (for entities) and x_3 (for time periods).

Another interesting question is whether common explainable artificial intelligence (XAI) methods, such as permutation-based feature importance, are also biased by the existence of shortcuts. Table 6 displays the deterioration in R^2 when randomly shuffling the values of predictors x_1, x_2, and x_3. Looking at the Lasso model, one could infer that it relies strongly on x_2. Looking at the Random Forest, one could infer that it mainly relies on x_2 and x_3. As according to the data-generation process described in Equation 1 neither x_2 nor x_3 are actually related to y, the results suggest that explainability methods such as permutation-based feature importance are also affected by shortcut learning bias.

To summarize, the results of our numerical analysis using simulated data strongly suggest that ML models exploit unobserved entity and time clusters as shortcuts, leading to overoptimistic performance evaluations and biased explainability metrics.

Table 3 **Permutation importance for variables x_1, x_2, and x_3.**

Method	$\Delta R^2(x_1)$	$\Delta R^2(x_2)$	$\Delta R^2(x_3)$
Lasso	0.010	0.073	0.010
Random Forest	0.040	0.176	0.236

3.3. Case Study: Energy Consumption Prediction

The ASHRAE Great Energy Prediction Dataset, which was published by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), presents a comprehensive view of energy consumption, architectural characteristics, and meteorological conditions of 1,400 structures across diverse regions of the United States (Miller et al. 2020). This dataset acts as an instrumental resource for the detailed examination of energy utilization trends, the formulation of strategies aimed at enhancing energy efficiency, and the projection of prospective energy conservation.

3.3.1. Overview of the Dataset and Methods The dataset provides extensive and detailed information about the various structures, which includes the buildings' geographical information, function or utility, spatial dimension (measured in square feet), construction year, and number of floors. Secondly, the energy usage component of the dataset provides granular energy consumption data for each of 1,000+ buildings and their respective metering devices. This consumption data is denoted in kilowatt-hours (kWh) and indexed by a timestamp as well as the meter type. Lastly, the dataset includes a weather conditions component, which furnishes daily meteorological data for each location represented in the dataset. This weather data encapsulates variables such as ambient temperature, cloud cover, precipitation, atmospheric pressure, wind velocity, and wind direction.

In order to have more stable estimates than in individual test runs, we conducted multiple cross-validation iterations. For each of the five cross-validation iterations, our training dataset – i.e., i.i.d. – comprised 100,000 samples while all test sets – i.e., i.i.d. and o.o.d. – comprised 20,000 unseen

samples. Given that our dataset spanned from January 1, 2016, to December 31, 2016 – i.e., a full year – we defined November 1, 2016, as our temporal cutoff point, hence, resulting in a temporal split that not only adheres to an 80/20 split but that also coincides with the beginning of the winter months. We defined the feature `site_id` as our entity feature for the entity-based sampling strategies.

We further used a comprehensive set of machine learning approaches.

1. a generalized linear baseline – i.e., a Lasso approach;
2. a Random Forest approach;
3. an eXtreme Gradient Boosting (XGBoost) approach (Chen and Guestrin 2016);
4. a LightGBM approach (Ke et al. 2017) – i.e., a gradient-boosting approach where trees are grown vertically as opposed to horizontally;
5. a TabNet approach (Arik and Pfister 2021) – i.e., a popular deep learning architecture specifically designed for tabular data; and
6. an Auto-ML approach based on the Scikit-Learn module (Pedregosa et al. 2011).

3.3.2. Results As can be observed in Table 4, it is primarily evident that aside from our benchmarks – i.e., a Lasso approach – there happens to be a marked deterioration in performance when any data sampling strategy other than a random one is utilized. This holds true for all implemented algorithms. Specifically, sampling a dataset based on a specific entity, consequently instigating a substantial shift in distribution, escalates the deterioration of all models in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Square Error (RMSE). This decline is remarkable, considering that the problem, the dataset, and the methodologies remained constant. Yet, in many academic scenarios, results would typically be reported based on a random sampling approach, leading, as would be the case here, to excessively optimistic performance estimates.

It is intriguing to note the relative stability of the Lasso approach showing minimal variance compared to more advanced methods. Despite lagging behind other methodologies in terms of performance on i.i.d. (test) sets, it outperformed most other approaches when dealing with o.o.d. samples. This implies that more straightforward procedures, perhaps due to their less complex nature, have a lower susceptibility to changes in distribution, thus providing a more robust solution in certain situations. In addition, TabNet shows favorable results in i.i.d. data but also only a limited deterioration in o.o.d. data which hints towards it also being a particularly good choice.

4. Addressing Shortcut Learning Bias: A Comprehensive Framework for Remediation

4.1. A Framework to Mitigate Shortcut Learning Bias

After identifying the problem of shortcuts, we describe a framework to cope with the problem in this section. The design of this framework is based on the observation that shortcut learning could be very common also in tabular data (as is the case in images and text, Geirhos et al. 2020) which makes many applications of machine learning in research susceptible to the problem. We suggest a three-step procedure in which the first step considers creating an i.i.d. distributed dataset without confounding observed features, the second step aims to detect potential shortcuts based on clustering (unobserved confounding features/clustering), and the third step suggests actions to alleviate or remove the problems resulting from clustering. The framework is mainly based on arguments of how to deal with confounding in machine learning (Schölkopf 2022) and unobserved characteristics in traditional statistical learning (Wooldridge 2010).

4.1.1. Data Quality Insurance and Study Design One problem that Geirhos et al. (2020) highlights is that while research is usually discussing test set performance, the actual interest will rather be the performance that will be achieved on new cases. It is common practice to assume that later new cases will be similar to the randomly drawn test cases, which is why this is not often

Table 4 Energy consumption forecast results

	Sampling Strategy	Evaluation		
		MAE	$RMSE$	R^2
Lasso	i.i.d. (test)	134.804	269.398	0.346
	o.o.d. (time)	135.925	249.094	0.34
	o.o.d. (entity)	160.287	276.921	0.328
	o.o.d. (time/entity)	153.105	266.671	0.286
Random Forest	i.i.d. (test)	69.269	134.142	0.836
	o.o.d. (time)	79.061	164.23	0.712
	o.o.d. (entity)	207.198	540.684	-2.415
	o.o.d. (time/entity)	184.67	484.451	-2.241
XGBoost	i.i.d. (test)	51.159	123.113	0.857
	o.o.d. (time)	62.85	150.637	0.755
	o.o.d. (entity)	200.501	485.483	-1.472
	o.o.d. (time/entity)	179.119	427.171	-1.272
LightGBM	i.i.d. (test)	57.468	125.066	0.853
	o.o.d. (time)	67.217	148.939	0.76
	o.o.d. (entity)	204.42	449.055	-1.075
	o.o.d. (time/entity)	181.72	402.759	-0.984
TabNet	i.i.d. (test)	96.112	204.009	0.622
	o.o.d. (time)	102.959	211.857	0.522
	o.o.d. (entity)	153.56	328.34	0.009
	o.o.d. (time/entity)	141.662	308.009	0.002
Auto-ML	i.i.d. (test)	35.273	87.11	0.931
	o.o.d. (time)	41.789	104.519	0.88
	o.o.d. (entity)	166.416	373.962	-0.355
	o.o.d. (time/entity)	156.826	346.036	-0.295

discussed in machine learning research. However, for really achieving similar performance as 'in the lab', one needs to make an honest assessment whether the data one is using is representative of later use cases. If data is not representative of the later use case, it will generally be difficult to train a reliable model. This is sometimes referred to as training distribution bias (Bueff et al. 2022).

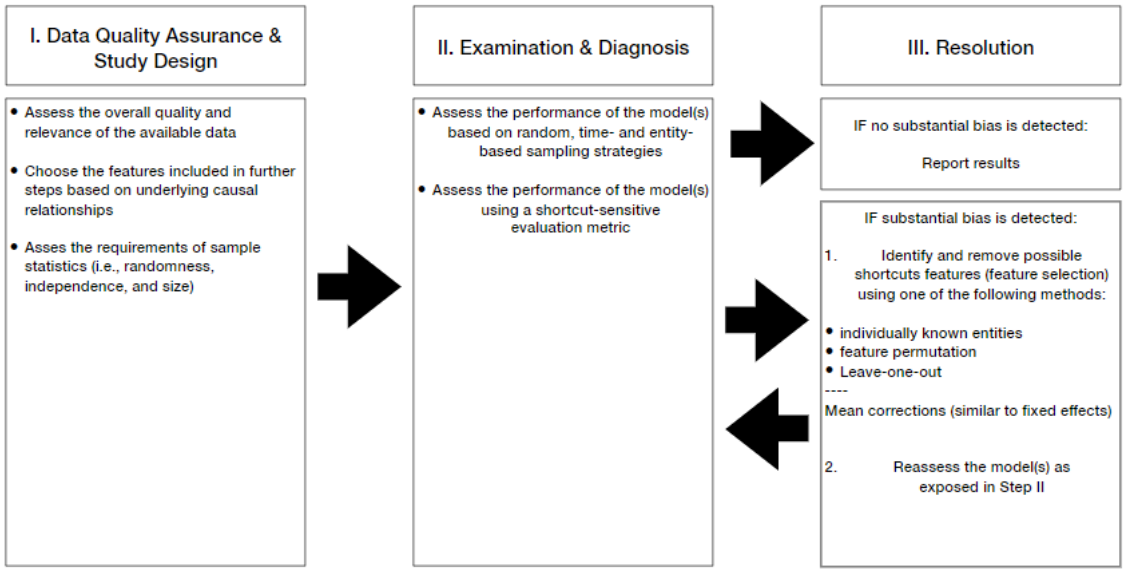

Figure 4 Framework for mitigating shortcut issues.

However, the discussion on i.i.d. and o.o.d. data sets acknowledges that new data will often not follow the exact same data generating process as i.i.d. data. Researchers and practitioners should therefore include actions to allow for generalizability. Considering the arguments on confounding as in Schölkopf (2022), one action to generalization is to reflect causal relationships in models and include features accordingly. This aims at preventing learning functional relationships to features that will not extend to new data with widespread examples such as in Alcorn et al. (2019) or Beery et al. (2018).

This paper mainly discusses bias in models by pooling different datasets. In one focal earlier described example, pneumonia is not detected via the x-ray images itself, but by identifying hospitals those images come from. This problem would, of course, not appear if the study design used truly i.i.d. data without any clusters that are later pooled. If this is not the case, a model can be subject to possible shortcuts due to bias from pooling data. However, there are remedies in this case that will be described in the next steps.

4.1.2. Examination and Diagnosis If there is reasonable suspicion that cases are not independent and differ in unobserved characteristics, it is worth assessing whether this results in bias of models and performance metrics. Besides straightforward clusters such as entities and time periods, the subsets could come from other clusters such as geographical regions.

We suggest two approaches to check for shortcut problems. The first one is to directly create synthetic o.o.d. data from the available data and assess the performance metrics on this data. This is the approach that was also used in Table 1 to detect shortcuts and was discussed earlier in the paper. If the performance metrics deteriorate compared to a simple i.i.d. test sample, this indicates that shortcuts are present.

For the second suggestion, researchers and practitioners could assess which level of performance could already be achieved by building a model of entities, years, or other clusters as features, respectively. If this model already achieves a performance that is similar to the level of the i.i.d. test performance or above, this is some indication of potential shortcuts. The advantage of this approach compared to the o.o.d. sampling lies in the fact that the models do not need to be refit for this procedure, which could be computationally intensive when checking many potential cross-sectional groups. This is indicated by the following equation:

$$P(f(X), D) \,\Theta\, P(g(entity, year), D) \tag{5}$$

where f is the functional relationship fit on the X features, g is a functional relationship fit on the entity and year, P is a performance metric, D is the respective test data, and Θ is some operator of comparison.

4.1.3. Resolution Based on the results from the shortcut detection, we suggest several actions to mitigate shortcut problems. These actions could be repeated until the metrics in the second step do not indicate shortcuts anymore. In this way, models could usually be used already in case there

are no indications of shortcuts. The solution aims to remove bias in estimators from unobserved variables. This bias results from the following circumstances (see Wooldridge 2010 for a discussion in traditional econometrics). The causal structure has the following form:

$$Y_{ci} = h(X_{ci}) + U_c + \epsilon_{ci} \tag{6}$$

where h is the true functional relationship, U_c are unobserved variables of cluster c (the clusters consist of combinations of the entity and the time period), ϵ_{ci} is some of i.i.d. error term, and i is an index for different cases of the same cluster. However, if U_c correlates with some variables of X_{ci} and one estimates a functional relationship f neglecting U_c, there will be a bias in the model in the following form.

$$Bias(f(X_{ci})) = E[f(X_{ci})] - E[Y_{ci}|X_{ci}] \neq 0 \tag{7}$$

where the expectation of the estimator $E[f(X_{ci})]$ will differ from the true expected value $E[X_{ci}|Y_{ci}]$ of Y_{ci} conditional on X_{ci}.

Practically, there are several potential options. The first option could lie in feature selection. In cases when only a small number of features in X_{ci} is correlated with U_c, there could be a trade-off to remove these features in order to remove the bias in the model. Researchers and practitioners could then trade a weaker i.i.d. test performance for a more robust o.o.d. performance. However, this approach requires a very clear understanding which variables could correlate with the unobserved variables besides a willingness to remove them from the model.

As a second remedy, we suggest following a similar procedure as is used by demeaning in fixed effects models in panel data econometrics (Wooldridge 2010).

$$X^*_{ci} = X_{ci} - \bar{X}_c, \quad Y^*_{ci} = Y_{ci} - \bar{Y}_c \tag{8}$$

where $\bar{X}_c$ are the means of features by cluster and $\bar{Y}_c$ are the means of targets by cluster. As this removes the unobserved variables, the estimated relationship is then unbiased for the i.i.d. data:

$$Bias(f^*(X^*_{ci})) = E[f^*(X^*_{ci})] - E[Y^*_{ci}|X^*_{ci}] = 0 \tag{9}$$

The unbiased estimates of the target could then be retrieved by adding the mean target in the cluster derived from the training data to $f^*(X^*_{ci})$:

$$prediction(X^*_{ci}) = f^*(X^*_{ci}) + \bar{Y_c} \tag{10}$$

One then needs to store some memory $\bar{X_1}, \bar{X_2}, ..., \bar{X_C}$ and $\bar{Y_1}, \bar{Y_2}, ..., \bar{Y_C}$ of the train mean features and targets in the C clusters. While this solves the bias on the i.i.d. data, where the cluster (combinations of years and entities) will be known, the clusters are unknown in the o.o.d. data.

The actions that are available, therefore, depend on the specific situation: (1) Where the entity means over multiple years from the past are the crucial part of the variation in U_c, initializing new values in the memory for future years with the mean over the past values of the entity will improve the performance over neglecting the clusters. (2) In many applications, it will be possible to acquire information on the cluster or even sample a small amount of cases for a new cluster. It is then straightforward to calculate the feature and target means in the cluster. (3) In other applications, it will be possible to have some sequential processing of cases. The estimates for mean features and targets in a new cluster could then be learned in a reinforced learning way. (4) If no information on the mean features and targets that could be expected in new clusters could be acquired. The means in the training sample should be used in making predictions. In this way, a biased model does at least not further add to the general uncertainty of forecasts. The uncertainty in the model should then be accounted for in decision-making.

4.2. Illustration of the Shortcut Learning Bias Mitigation

In this section, we show how to address the shortcut learning problems in the simulated data as used earlier. The earlier analyses have detected problems of shortcuts as indicated by a deterioration in

Table 5 Simulation results on shortcut mitigation

	Sampling Strategy	MAE	$RMSE$	R^2
	No further information:			
Lasso	i.i.d. (test)	0.796	0.997	0.992
Lasso	o.o.d. (time)	7.341	8.892	0.358
Lasso	o.o.d. (entity)	11.317	13.322	0.166
Lasso	o.o.d. (time/entity)	16.170	18.782	-0.158
Random Forest	i.i.d. (test)	0.001	0.009	1.000
Random Forest	o.o.d. (time)	7.268	8.842	0.365
Random Forest	o.o.d. (entity)	9.891	12.708	0.241
Random Forest	o.o.d. (time/entity)	15.319	18.240	-0.092
	Group information acquirable:			
Lasso	i.i.d. (test)	0.796	0.997	0.992
Lasso	o.o.d. (time)	0.841	1.058	0.991
Lasso	o.o.d. (entity)	0.845	1.056	0.995
Lasso	o.o.d. (time/entity)	0.842	1.049	0.996
Random Forest	i.i.d. (test)	0.001	0.009	1.000
Random Forest	o.o.d. (time)	0.002	0.020	1.000
Random Forest	o.o.d. (entity)	0.002	0.017	1.000
Random Forest	o.o.d. (time/entity)	0.002	0.018	1.000

performance for moving from i.i.d. metrics to o.o.d. metrics. We, therefore, start by demeaning the training features and targets within all combinations of the entity and time period in the training data (the means are stored as a memory). We fit a clean model afterward and make a prediction for the test data by first deducting the training cluster means for the features and then later adding the training cluster mean of the target to the prediction. The resulting performance metrics are displayed in the first line of each section in Table 5. One can see that after the bias is removed, the model captures the underlying functional relationship almost perfectly, as indicated by low MAE and $RMSE$ and an R^2 above 0.9. Further assessing Table 6, one can see that the variable

importance now gives evidence that only variable x_1 is important in both models, which correctly reflects the causal structure of the data-generating process.

We then discuss the prediction performance on the o.o.d. samples. This is presented in the following lines in Table 5. We make several observations. The first section of the table contains results where no further information on the clusters could be acquired and means over entities (o.o.d. time), time periods (o.o.d. entity), or the whole training sample (o.o.d. time/entity) are used. One could see from the table that the performance is stronger compared to Table 2. Furthermore, the more complex model remains superior to the more simple model. The second section of Table 5 then presents results, where information could be acquired by sampling a small number of cases from each new cluster for learning the quality of the cluster (i.e., having access to prior electricity bills). After including the derived means, the functional relationships are captured about as good as in the i.i.d. sample.

Table 6 Permutation importance for variables x_1, x_2, and x_3 after shortcut mitigation

Method	$\Delta R^2(x_1)$	$\Delta R^2(x_2)$	$\Delta R^2(x_3)$
Linear model	0.009	0.000	0.000
Random Forest	0.009	0.000	0.000

5. Conclusion

This paper studies the prevalence of shortcut learning in tabular data. Shortcut learning has previously been widely found in unstructured data such as text or images, but has not been discussed in tabular data to the best of our knowledge. We discuss how shortcuts appear in theory and show that shortcuts easily appear in simulated data and in energy consumption prediction. Given that studies in a literature review mostly rely on random sampling, the problem is likely widespread. While many studies might not suffer from the problem, we aim to start a discussion

about where shortcuts might have influenced results in the past. We further suggest discussing measures of potential shortcut problems in future work.

References

Alcorn MA, Li Q, Gong Z, Wang C, Mai L, Ku WS, Nguyen A (2019) Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4845–4854.

Arik SÖ, Pfister T (2021) TabNet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6679–6687.

Beery S, Van Horn G, Perona P (2018) Recognition in terra incognita. *Proceedings of the European conference on computer vision (ECCV)*, 456–473.

Bueff A, Papantonis I, Simkute A, Belle V (2022) Explainability in machine learning: a pedagogical perspective. *arXiv* 2202.10335.

Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chollet F (2019) On the measure of intelligence. *arXiv* 1911.01547.

Friedman J, Hastie T, Tibshirani R (2009) *The Elements of Statistical Learning* (Springer).

Geirhos R, Jacobsen JH, Michaelis C, Zemel R, Brendel W, Bethge M, Wichmann FA (2020) Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2(11):665–673.

Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W (2018) ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv* 1811.12231.

Geva M, Goldberg Y, Berant J (2019) Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *arXiv* 1908.07898.

Hyndman RJ, Athanasopoulos G (2018) *Forecasting: Principles and Practice* (OTexts).

Kaufman S, Rosset S, Perlich C, Stitelman O (2012) Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data* 6(4):1–21.

Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 30.

Miller C, Arjunan P, Kathirgamanathan A, Fu C, Roth J, Park JY, Balbach C, Gowri K, Nagy Z, Fontanini AD, Haberl J (2020) The ASHRAE great energy predictor III competition: Overview and results. *arXiv* 2007.06933.

Ng AY (1997) Preventing overfitting of cross-validation data. *International Conference on Machine Learning*, 245–253.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. (2011) Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research* 12:2825–2830.

Rosenfeld A, Zemel R, Tsotsos JK (2018) The elephant in the room. *arXiv* 1808.03305.

Schölkopf B (2022) Causality for machine learning. *Probabilistic and Causal Inference: The Works of Judea Pearl*, 765–804.

Wooldridge JM (2010) *Econometric Analysis of Cross Section and Panel Data* (MIT Press).

Zhu Z, Xie L, Yuille AL (2017) Object recognition with and without objects. *arXiv* 1611.06596.