Causal Feature Selection for Fairness and Performance in EDM Predictions: Concerns About Causal Modeling Should Not Stop Us

Lea Cohausz
University of Mannheim
Mannheim, Germany
lea.cohausz@uni-mannheim.de

Recent work in educational data mining (EDM) emphasizes the benefits of using causal methods in machine learning (ML) across various tasks. One task that lends itself particularly to causal approaches is that of feature selection (FS). Causal feature selection (CFS) methods theoretically yield optimal feature subsets, enabling efficient ML models, interpretability, and insights for interventions. Moreover, CFS can easily be extended to incorporate fairness constraints, another important topic in EDM. Yet, CFS and other causal methods have rarely been used in practice so far because they require causal graphs, which are challenging to construct with observational data. With no ground truth available to compare to, the true accuracy of the constructed causal graphs is hard to gauge. In this paper, we investigate, using five educational datasets, how CFS methods compare with other FS methods and other fair pre-processing methods, examining the predictive performance and algorithmic bias of the resulting ML models. By considering the results of CFS methods on downstream tasks, we not only show that these methods offer competitive results on some datasets (although a specific branch of FS methods typically outperforms them regarding predictive performance), but by doing so, also provide an indication that the constructed causal models were able to uncover relevant relationships. This finding may make us more optimistic about using causal methods in practice. The code relevant for our experiments is available here.

Keywords: causal feature selection, algorithmic fairness, causal modeling, directed acyclic graphs

1. Introduction

Feature selection (FS)—the process of choosing a subset of features from a complete dataset to predict a target variable—is a key step in machine learning (ML) (Batool et al., 2023). Especially in specific settings, such as often observed in education, tabular datasets contain relatively few samples (e.g., two widely used public educational datasets include only 480 and 131 students, respectively (Amrieh et al., 2016; Hussain, 2018)) but a moderate or even large number of features. A high feature-to-sample ratio makes it difficult for ML models to identify general patterns and limits predictive performance. Using fewer features not only improves predictive performance but also efficiency and can enhance model interpretability, an advantage particu-

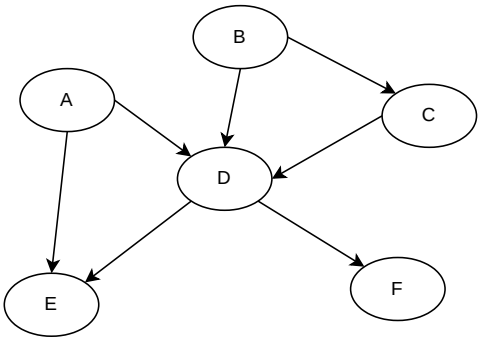

Figure 1: An example of a DAG.

larly helpful for the high-stakes area of education (Batool et al., 2023; Cohausz, 2025).

One specific branch of FS that can provide additional benefits is that of causal feature selection (CFS). CFS selects features based on a graphical model of the data-generating mechanism of a dataset. This causal graph is typically modeled as a directed acyclic graph (DAG), in which nodes represent variables[1] and directed edges the causal relationships (Guyon et al., 2007; Yu et al., 2020). Figure 1 shows a very simple example DAG. When the DAG is specified correctly, the selected features contain all the information needed to train an ML model without loss of information, leading to a highly efficient and accurate model (Yu et al., 2021). In addition, a DAG can give additional insights, such as relationships among features that can guide potential interventions when predicting, e.g., struggling students (Cohausz, 2025). CFS also provides a straightforward approach to achieving algorithmic fairness. Algorithmic fairness is typically defined as an independence between one or several sensitive features (e.g., gender) and the target variable—the existence of such a statistical dependence is also called an algorithmic bias. Kusner et al. (Kusner et al., 2017) proved that when those features are excluded from a DAG that are descendants of the sensitive feature, the resulting model trained with the remaining features is fair.

Because of the advantages of CFS, there has been a call in educational research to focus on these methods (Cohausz, 2025). More generally, the call for using CFS in education can be seen as a part of a wider movement in EDM and related areas that call for explicit causal modeling for a variety of reasons (Cohausz, 2025; Weidlich et al., 2024). However, a major hurdle in using causal methods—as also highlighted by the papers calling for causal modeling—is the requirement to construct causal DAGs, which is a non-trivial and error-prone task (Cohausz et al., 2025). Although there have been some advances in constructing DAGs recently, it is still unlikely that DAGs will be constructed perfectly and represents true causal relationships (Cohausz et al., 2025). Because there is no ground-truth DAG to compare against when using real (i.e., non-synthetic) data, we can never be certain whether or to what extent a DAG has been learned correctly. This problem is likely also the reason why a practical evaluation of how well these methods actually work is missing: CFS methods have not been systematically tested on real data, and most claims about their benefits remain theoretical (Ling et al., 2024; Saarela, 2024).

[1]The terms variable and feature are used interchangeably in this work.

However, the downstream performance of CFS methods, in terms of predictive performance and algorithmic fairness, relative to other methods provides an indication of how correctly these DAGs are constructed, as no other method can theoretically perform better than the causal methods if DAGs are learned correctly (Yu et al., 2021). If CFS methods perform reasonably well on downstream tasks, this finding would have two consequences: First, it would indicate that the constructed DAG provides a relatively good model of the data-generating mechanism. Second, despite the DAG being unlikely to be perfectly accurate, it still provides useful information and results, highlighting its value for the mentioned tasks (performance, fairness) and beyond. Both of these aspects would weaken arguments against the use of causal methods that are based on the difficulty of achieving accurate models, meaning that the models may be unusable for subsequent tasks, and on the difficulty of assessing the accuracy. This paper is therefore concerned with an empirical evaluation of how CFS methods compare with non-causal FS methods on predictive performance, and with how CFS methods with fairness constraints compare with other pre-processing fairness methods on algorithmic bias using educational datasets. At the same time, this evaluation provides a more general framework for assessing how useful the learned DAGs are and how close they are to the (unknown) ground truth by examining downstream tasks. Our concrete evaluation uses five publicly available, diverse tabular educational datasets. We evaluate a range of ML and FS methods with respect to predictive performance and algorithmic fairness. Specifically, we address the following research questions (RQs):

RQ1: Do ML models trained in the feature subset selected via CFS methods outperform the baseline of using all features?

RQ2: How well do ML models trained on features selected via CFS perform compared to other models trained with features selected via other FS methods?

RQ3: How well do fair CFS methods reduce algorithmic bias compared to other fair pre-processing methods?

RQ4: How do fair CFS methods compare to other fair pre-processing methods with respect to the trade-off of predictive performance and reducing algorithmic bias?

In general, the RQs determine whether the constructed DAGs identify truly relevant relationships to such an extent that they are competitive. The remainder of the chapter is organized as follows:

- In Section 2, we introduce the area of (causal) feature selection for predictive performance and the basics of causal modeling. Subsequently, we discuss algorithmic fairness and how to mitigate it with fair CFS methods in Section 3.
- In Section 4, we explain how we evaluate the research questions and introduce our evaluation pipeline, the FS and fair pre-processing methods we compare against, the datasets, and the evaluation metrics. In Section 5 we then present the results.
- In Section 6, we conclude the paper, highlighting limitations and pointing to future research.

2. Feature Selection for Predictive Performance

First, we briefly introduce (causal) feature selection for predictive performance.

2.1. Feature Selection

The motivation for applying FS when training an ML model is that not all features $\mathbf{X}$ are relevant for predicting a target variable Y (Yu et al., 2021). Some features may provide no useful information about the target variable, while others may only duplicate information already captured by other variables. By selecting a subset of informative features, we help the model focus on the important features, improving generalization and computational efficiency (Yu et al., 2020). In this paper, we denote the final set of features used to train the model as $\mathbf{T}$. FS methods are usually grouped into two main families: filter and wrapper methods (Yu et al., 2021; Wah et al., 2018). Filter methods rank features by their association with the target variable (e.g., via correlation) and select the top-ranked features (Yu et al., 2021; Wah et al., 2018). Wrapper methods evaluate subsets of features by training the model directly and measuring how well those subsets predict Y; features are added or removed iteratively (Yu et al., 2021; Wah et al., 2018). Because the ML algorithm itself is used during selection, the chosen model influences which features are retained. Causal feature selection (CFS) is formally a type of filter method, since it does not rely on a trained model.

2.2. Causal Feature Selection

Causal feature selection (CFS) is based on the graphical model of the data-generating mechanism of a specific setting. As already explained, the graphical model is typically a causal DAG, and the DAG encodes (conditional) independence relationships among the variables represented by nodes in the DAG. Most importantly for CFS, any variable V is independent of all other variables given its parents (all variables in the graph connected to V with the arrow pointing towards V), its children (all variables in the graph connected to V with the arrow pointing away from V), and its children's other parents (also called its spouses). This set of variables is referred to as the Markov blanket (MB) of V. In our example DAG in Figure 1, the MB of node D consists of parents A, B, and C and children E and F. The MB of A consists of child E and spouse D.

For CFS, this means that the MB of the target variable Y, denoted as $MB(Y)$, encodes all relevant information concerning Y and no other variable adds any additional information beyond it; the set of retrieved features is minimal (Pearl, 2009). This makes models trained with features selected by CFS particularly efficient, as even highly correlated features may not be selected if all the information they contain is already entailed in other features. Using information theory, Yu et al. (Yu et al., 2021) show that causal and non-causal FS share the same objective and that a correct MB is the optimal (and therefore also minimal) feature subset. The main challenge in CFS is therefore to construct a DAG (after which identifying the MB is trivial) or to learn the MB directly.

When only observational data are available (in contrast to experimental data, which is often impossible to obtain), this is a difficult task. Generally, DAGs can be constructed using background knowledge, data-driven structure learning methods, or a combination of both (Cohausz et al., 2025). Typically, some background knowledge is available, e.g., we may know that certain relationships must or cannot exist. However, background knowledge is often not detailed enough to specify a full DAG (Weidlich et al., 2024; Cohausz et al., 2025), and for the datasets in this study, we have very limited background information available. Data-driven structure learning can be divided into two families of methods: constraint-based and score-based structure learning (Kitson et al., 2023). Very briefly put, constraint-based methods use (conditional) independence tests to identify an undirected graph (by removing edges between conditionally

independent variables from an initially fully connected graph). Then, some edges can be oriented by identifying so-called v-structures. These v-structures consist of three nodes connected by two edges (A—B—C) for which it holds that the middle node is not part of the set of variables that make the other two nodes independent. These v-structures can then be oriented as follows $A \rightarrow B \leftarrow C$. Based on these orientations, other edges can be oriented to avoid new v-structures or cycles (Kitson et al., 2023). A special variant of the constraint-based methods are the local structure discovery methods. Local structure discovery focuses on each variable's MB or on its parents and children using conditional independence tests—if FS is the sole goal, having the MB of Y can be enough and efficient. If a DAG is needed (e.g., for fairness), it can then be assembled from these local relationships (Kitson et al., 2023). Score-based methods, in contrast, iteratively add or delete edges that increase a score the most in an initially unconnected graph. The score represents how close the graph fits to the data (Kitson et al., 2023). Both families of methods have many different algorithms but are known to be error-prone[2]. A problem with DAGs learned purely from data is that they need not necessarily represent causal relationships. Score-based methods cannot make any guarantees with regard to recovering truly causal relationships. Constraint-based methods can do give such guarantees but only under specific assumptions. Background knowledge that encodes relevant causal information can help with making these DAGs more causal.

Recently, research has shown that combining background knowledge and structure learning leads to improved results with regards to uncovering the true causal DAG, with Cohausz et al. showing that even small restrictions in the search space or in edge orientation based on the knowledge we implicitly have when concerned with algorithmic fairness lead to sometimes large improvements (Cohausz et al., 2025). Because, as stated, we lack more detailed background knowledge of the datasets in this study, we follow, whenever possible, the strategy established by Cohausz et al., which incorporates these minimal assumptions: (1) demographic features (e.g., gender)—which are usually considered relevant for fairness—cannot be influenced by other features, as their values are fixed before the other features' values, and (2) the target variable cannot influence any other feature, as it is determined last in time (Cohausz et al., 2025). More background knowledge may help make the DAGs more accurate and causal, but for the aim of this study, it is not a weakness if the DAGs are not as accurate as possible, as we are interested in how imperfect DAGs affect subsequent steps. It should be highlighted that CFS (for predictive performance and fairness) only works when the DAG is the true causal DAG; therefore, any poor performance in downstream tasks can be attributed to the learned DAG deviating from the true causal DAG.

2.3. Structure Learning Methods

We will now briefly describe the DAG learning methods we use in this study. We put a particular emphasis on local methods, as they can be efficient for FS. As a global approach, we use the score-based Hill Climber (HC) algorithm, which has repeatedly been found to perform well (Cohausz et al., 2025). It just adds or deletes the edge that leads to the biggest score increase (Cohausz et al., 2025). When we extract the MB from the DAG learned by HC, we refer to the method as HC-MB. As one local method for learning complete DAGs, we use the constraint-based HITON-Parents-Children (HITON-PC) algorithm. Unlike HC's global search,

[2]For more information on the families of methods, concrete methods and their problems, please consult: (Kitson et al., 2023).

HITON-PC first constructs a skeleton by identifying each variable's parents and children through conditional independence tests, and then orients edges to form a full DAG (Kitson et al., 2023). Parents–children approaches often perform well (Kitson et al., 2023), but from an FS perspective, they require learning the full DAG from each node's parents and children, because the spouses are missing otherwise. In addition, we employ the Incremental Association Markov Blanket (IAMB) algorithm, another local method that directly learns MBs. IAMB is especially suitable when the goal is to find only the target's MB and is known to scale well to large numbers of variables, a setting where FS benefits most (Kitson et al., 2023). Finally, we include LRH (lessen swamping, resist masking, and highlight the true positives), a local algorithm designed to mitigate errors in conditional independence tests. Its authors report strong performance across a range of conditions (Liu and Liu, 2016).

2.4. Related Work

A substantial body of research has examined FS on general ML or educational datasets, with several studies in the educational domain highlighting not only gains in performance and efficiency but also the potential for deeper insights (Zaffar et al., 2021; Bertolini et al., 2021). Although these works also have some limitations with regard to the generalization of their results (e.g., it is often not investigated whether FS even improves the predictive performance compared to using the complete feature set, especially for tree-based methods (Ramaswami and Bhaskaran, 2009; Bertolini et al., 2021)), the body of work comparing non-causal FS methods is large. The same cannot be said for CFS methods. Despite being a well-established and active subfield of FS (Guyon et al., 2007; Yu et al., 2021), CFS has received virtually no attention in EDM (Cohausz, 2025), but also beyond (Chandrashekar and Sahin, 2014; Jović et al., 2015). Most work on CFS focuses on discovering MBs when data is missing, non-linear, or high-dimensional (Yu et al., 2022; Ling et al., 2022), or when conditional independence tests fail (Guo et al., 2022), and is evaluated only on synthetic datasets. An evaluation of downstream tasks and using real-life data is almost never considered. A notable exception is Saarela (Saarela, 2024), who showed that CFS performed slightly worse than correlation-based FS for specific settings. More generally, Yu et al. (Yu et al., 2021) found on synthetic and standard ML datasets that causal and non-causal FS achieve comparable predictive performance. When ample training data are available, CFS can even outperform non-causal approaches. Yet, with small sample sizes, correlation-based methods tend to perform better, likely because causal structure-learning methods incorrectly fail to insert edges between truly related variables. However, overall, an evaluation using multiple datasets, multiple ML, and non-causal FS methods is still missing.

3. (Causal) Feature Selection for Algorithmic Fairness

We will now discuss how we can use these methods to mitigate algorithmic bias. Before we do so, we first need to discuss how we define algorithmic bias in this paper.

3.1. Algorithmic Bias

Generally, selecting an appropriate notion of algorithmic bias is non-trivial and depends on factors such as the application context (Cohausz et al., 2025). In this study, we have a different focus: we analyze how using fair CFS and other fair pre-processing methods affects standard measures of algorithmic bias. We do not claim that the resulting models are universally "fair".

Indeed, because the evaluation happens without a specific application scenario in mind and without deeper knowledge of the context, we cannot make such claims, and it is also not necessary for the evaluation done here. We evaluate how these methods influence commonly used algorithmic bias metrics—concretely, the group algorithmic bias metrics of demographic parity (DP) and equalized odds (EO) (Hardt et al., 2016; Dwork et al., 2012). DP encodes the fairness idea that predictions $\hat{Y}$ should be independent of the sensitive feature S. It requires that the positive prediction rate is identical across all sensitive groups (Dwork et al., 2012), i.e., in the binary case:

$$P(\hat{Y} = 1|S = 0) = P(\hat{Y} = 1|S = 1) \tag{1}$$

EO takes the ground truth Y into account, assumes that it is unbiased, and encodes the notion that both false positive and false negative rates across the sensitive groups should be equal (Hardt et al., 2016), i.e.:

$$P(\hat{Y} = 1|S = 0, Y = y) = P(\hat{Y} = 1|S = 1, Y = y) \tag{2}$$

Throughout this study, we work with the demographic parity ratio (DPR) and equalized odds ratio (EOR). The ratios are calculated by dividing the smallest rate by the largest rate. For example, if $P(\hat{Y} = 1|S = 0) = 0.4$ and $P(\hat{Y} = 1|S = 1) = 0.7$, then we divide $\frac{0.4}{0.7}$. For both DPR and EOR, a model is considered fair according to the respective notion if the rate is close to 1. An often-cited rule of thumb is that models are considered fair if the ratio is > 0.8.

3.2. Fair CFS Methods

Kusner et al. demonstrated that removing all variables from the training data that are either sensitive or descendants of sensitive variables in a DAG yields a model that satisfies standard measures of algorithmic bias (Kusner et al., 2017). Hence, the first approach to learning fair models we include in this study is to exclude all descendants of the sensitive variable and the sensitive variable. Hence: $\mathbf{T} = \mathbf{X} \setminus \mathbf{D}$ where $\mathbf{D}$ are the descendants and the sensitive variable. The sensitive variable is excluded for all of the following methods. We abbreviate this method as ExDesc. ExDesc is solely focused on reducing biases; it is not supposed to improve predictive performance, as features are not selected based on their relationship with the target variable. This objective is different when compared to the second method to achieve a fair model, which is to further restrict the training data to those variables that are not descendants of the sensitive variable and are in the MB: $\mathbf{T} = MB(Y) \setminus \mathbf{D}$. We call this method MBExDesc. Note that it may be the case that no feature remains after excluding the descendants. In such a case, no model can be learned. Again, we use all CFS methods from above. MBExDesc and ExDesc require the complete DAGs.

In contrast, the FairCFS algorithm proposed by Ling et al. (Ling et al., 2024) only requires the MBs of the sensitive $MB(S)$ and the target $MB(Y)$ variable. Then, we select features according to two rules.

- If a feature $X_i \in \mathbf{X}$ is in the MB of the target but not in the MB of the sensitive variable (i.e., $X_i \in \{MB(Y) \setminus MB(S)\}$, then it will be added to $\mathbf{T}$ if and only if X_i is independent of S given the Markov Blanket of S ($X_i \perp S|MB(S)$).
- Alternatively, a feature X_i that is in both MBs (i.e., $X_i \in \{MB(Y) \cap MB(S)\}$) can be added to $\mathbf{T}$ if it is independent of S given a set $\mathbf{Z}$ that is part of the MB of S, i.e., $X_i \perp S|\mathbf{Z}, \mathbf{Z} \subseteq \{MB(S) \setminus \mathbf{X}\}$.

For the proof of why the resulting feature set $\mathbf{T}$ is unbiased, please consult Ling et al. (Ling et al., 2024). Again, it may be the case that no feature is available for training the model after these steps.

Our fourth causal method is a novel approach that we call causal de-correlation (CaDe). The goal is to preserve more non-problematic information from variables in $MB(Y)$ that are descendants of the sensitive variable S, which should better balance the trade-off between predictive performance and bias mitigation, and to make sure that this method can even be applied when all variables are descendants of the sensitive feature. Let $\mathbf{D}$ again denote the descendants of S and $MB(Y)$ the target's Markov blanket. For each feature $X_i \in \{\{MB(Y) \setminus S\} \cap \mathbf{D}\}$, we de-correlate X_i from S, producing $\hat{X}_i$, and replace X_i with $\hat{X}_i$ in the $MB(Y) \setminus S$ to form the final feature set $\mathbf{T}$. Note that all of the fairness strategies are used on the learned structures returned by all described structure learning methods.

3.3. Related Work

The idea of using CFS to create fairer ML models originated primarily in the causal fairness community (Kusner et al., 2017), starting with the work of Kusner et al. As we just discussed, subsequent research has proposed alternative strategies that retain more predictive power, such as methods that block the flow of information from sensitive variables rather than removing every affected variable (Ling et al., 2024). Other approaches avoid learning a complete DAG altogether (Galhotra et al., 2022), although we argue that this limits the advantages of causal methods. These causal bias mitigation methods are rarely compared with each other or with non-causal methods, and are usually evaluated on only one real-world dataset (Kusner et al., 2017). As stated in the introduction, the concept of fairness-oriented FS has also gained traction in the broader FS literature. Notably, Belitz et al. (Belitz et al., 2021) introduced a wrapper method that selects features based on both predictive performance and an algorithmic fairness metric; however, this was not compared against CFS methods. Apart from this method, however, most other fair pre-processing methods that have evolved as standard techniques are not based on feature selection but rather on removing the influence of sensitive features by adjusting the values of the affected features (Bellamy et al., 2019; Bird et al., 2020).

4. Evaluation Pipeline

In the following, we will discuss how we evaluated our RQs. First, we will discuss the choice of non-causal FS and fair pre-processing methods we compare against, then the general evaluation pipeline, the ML methods, including the hyperparameter settings, and the evaluation metrics. Subsequently, we will introduce the datasets.

4.1. Feature Selection Methods

As noted earlier, FS methods are commonly divided into filter and wrapper approaches, and our study compares techniques from both families against the CFS family. We chose only methods well-suited to classification tasks, as our evaluation is limited to classification to ensure comparability across experiments.

4.1.1. Filter Methods

Filter methods primarily differ in the scoring metrics they use to assess a feature's relevance to the target variable. Two widely used scores are ANOVA F-values (hereafter SelectF) and mutual information (MI) (Chandrashekar and Sahin, 2014), and we use both strategies. We selected the methods because they work well for classification tasks and impose few restrictions on the data. Both methods require a hyperparameter k, which specifies the number of features to select.

4.1.2. Wrapper Methods

As wrapper methods, which are generally more computationally demanding than filter methods, we include two approaches that operate independently of the downstream ML model (L1-based selection and Boruta) and two that are closely tailored to the chosen ML model (recursive feature elimination and sequential forward selection). L1-based selection (L1) leverages the properties of L1 regularization, which drives coefficients to exactly zero, leaving a set of strong predictors with nonzero weights (Schmidt, 2005). Boruta typically relies on random forest feature importance scores (Kursa et al., 2010). It generates shadow features by randomly shuffling real features (thus preserving their distributions) and compares the importance of the original features to these shadows. Features that remain consistently less informative than their shadows are discarded (Kursa et al., 2010).

Recursive feature elimination (RFE) and sequential forward selection (SFS) are also wrapper methods but rely on repeatedly training ML models on varying feature subsets and observing how predictive accuracy changes in response (Wah et al., 2018). RFE begins with all features and removes the least important ones determined step by step, whereas SFS starts with none and adds the most informative features iteratively. Unlike Boruta and L1, which automatically determine the number of features to retain, RFE and SFS require the user to specify k. Although RFE and SFS select features based on the performance of specific ML models (e.g., random forest) using these features, the resulting feature subsets can be used to train other ML algorithms (e.g., logistic regression) (Wah et al., 2018).

4.2. Fair Pre-Processing Methods

We compare the fair CFS methods with another non-causal FS fairness method and two standard fair pre-processing methods: disparate impact removal and correlation removal. The fair FS method, which we mentioned already, is called fairness-aware feature selection (FA-FS) (Belitz et al., 2021). It is based on sequential forward selection. In standard SFS, features are added based solely on their contribution to predictive accuracy. FA-FS adds a fairness component, selecting features that balance improvements in accuracy against increases in unfairness. The relative weight between accuracy and fairness is controlled by a hyperparameter, which we set to give equal importance to both objectives. The reason for including this method is that it also works via feature selection.

Disparate impact removal (*DR*) works by modifying feature values so that the outcomes of an ML model are less dependent on the sensitive features (Bellamy et al., 2019). The goal is usually to achieve demographic parity. To do so, DR modifies the input data, adjusting the distributions to reduce potential unfairness, while leaving the model unchanged. In contrast, the correlation remover method (*CR*) focuses on the relationship between regular and sensitive features rather than on outcomes (Bird et al., 2020). It works by removing linear correlations between the regular and sensitive features, effectively de-correlating them. This approach aims to prevent

the model from implicitly learning bias from the features themselves. DR and CR were chosen as comparisons because they are among the most well-known and established methods for fair-preprocessing but are also represent different types of fair pre-processing (focused on outcome and not focused on outcome) methods.

4.3. Overview

For each dataset, we employ 5-fold cross-validation. In each fold, we first run all ML methods without FS or fair pre-processing, followed by all possible combinations of FS/fair pre-processing and ML methods. We also evaluate model-specific wrapper methods with other ML algorithms (e.g., using features selected by FSF based on random forests as input to logistic regression).After computing the evaluation metrics for each fold, we average them across folds for each FS–ML method combination. We calculate all metrics for all combinations, including algorithmic bias metrics for FS methods that do not explicitly address fairness. Because some FS methods (particularly SFS, HITON-PC, and LRH) do not scale well, we implemented a 30-minute time-out. If an FS method did not finish within 30 minutes for a fold, it was terminated, and we proceeded with the next method. Random seeds are set to ensure reproducibility.

4.4. ML Methods and Hyperparameters

We use seven ML methods in our evaluation to cover a broad range of model types. Three of them are classical methods commonly employed in FS studies: logistic regression (LR), support vector machines (SVM), and k-nearest neighbor (kNN). We also include three tree-based methods: random forest (RF), eXtreme Gradient Boosting (XGB), and CatBoost. Tree-based methods continue to outperform other approaches on tabular data[3] (Erickson et al., 2025). Finally, we include a small feedforward neural network (NN) to represent basic deep learning approaches.

We do not perform hyperparameter optimization as our goal is not to achieve the best possible results but rather to gain general insights into emerging patterns with regard to the performance of CFS methods. Additionally, the computational load of our experiments is already substantial, with a total of 259 models evaluated per dataset. Hyperparameters are set to reasonable default values across all datasets rather than being tuned individually. Regarding the hyperparameter k, which is required by some FS methods and determines how many features to select, we run each method multiple times per fold with different values of k. Specifically, we set k to 20%, 40%, 60%, and 80% of the total number of features in the training dataset, respectively.

4.5. Pre-Processing

Similarly to our approach for hyperparameter selection, we apply a consistent pre-processing strategy across all datasets. All non-numeric features are recoded: binary non-numeric values are converted to 0 and 1, ordinal categorical features are mapped to numeric values, and non-ordinal categorical features are one-hot encoded. ID columns and columns without variance are removed.

[3] Although foundation models and some deep learning methods are closing the gap, they still have limitations regarding the number of features and samples, making them impractical for all datasets in this study (Erickson et al., 2025).

4.6. METRICS

As our primary performance metric, we use the macro F1 score, which is appropriate for datasets with imbalanced target classes. As already stated, we quantify algorithmic fairness using the DPR and EOR. For all datasets, we decided to only compute the fairness metric for the sensitive variable gender. The reason is that this variable is present in all datasets and is almost always considered sensitive in educational settings. Given that our experiments and, thus, the results are already extensive, we decided that considering only one variable is the best choice.

4.7. DATASETS

We use five datasets in our evaluation. The datasets are summarized in Table 1, which lists the number of samples and features after pre-processing, as well as the distributions of the target and gender variables. We selected the datasets by examining all datasets discussed by Mihaescu et al. (Mihaescu and Popescu, 2021) in their review on educational datasets. We included datasets that either represent a classification task and are in the tabular, stationary data domain (not sensor, image, or time-series data) or can be transformed to meet these criteria. Based on these requirements, we selected four datasets. We added a fifth dataset (early learning predictors) that was recently used in a data challenge on the data science competition platform Zindi and complements the other datasets, as it has a large feature space and focuses on a different kind of target variable. Details about the specific pre-processing can be found online.

4.7.1. xAPI Dataset

The students' academic performance (xAPI) dataset consists of 480 samples (Amrieh et al., 2016). The goal of the dataset is to assign school students (mostly from Kuwait and Jordan) to three categories of performance (low, medium, high) based on demographic and behavioral information. There are no missing values in the dataset. Roughly 64% of the students in the dataset are male, the remaining female.[4]

4.7.2. SAP

According to the authors, the Student Academics Performance (SAP) dataset consists of 300 samples, but only 131 are present in the publicly available dataset (Hussain, 2018). Like the xAPI dataset, the goal of this dataset is to accurately predict student performance. In this case, college students are categorized according to their end-of-semester percentage into four groups. There are no missing values in the dataset.

4.7.3. OULAD

The Open University Learning Analytics Dataset (OULAD) consists of information on adult learners taking online courses in the UK (Kuzilek et al., 2017). The dataset theoretically lends itself to many different prediction tasks. We follow other papers (e.g., (Cohausz et al., 2023)) and predict whether someone passes or fails their course. Pre-processing for this dataset is more complex, as several tables have to be merged, and some data (e.g., interaction with the virtual learning environment) needs to be summarized for this prediction task. There are only a few

[4]For this and all other datasets used here, other categories for gender are not reported.

Table 1: A brief description of the datasets.

Dataset	# Sample	# Features	Feature-Sample Ratio	Target Dist.	Gender Dist.
xAPI	480	39	0.081	0.26, 0.44, 0.3	0.64 (M), 0.36 (F)
SAP	131	37	0.282	0.21, 0.41, 0.32, 0.06	0.55 (M), 0.45 (F)
OULAD	21,189	32	0.002	0.69, 0.31	0.54 (M), 0.46 (F)
DAP	12,411	170	0.014	0.08, 0.16, 0.25, 0.51	0.59 (M), 0.41 (F)
ELP	8,475	437	0.052	0.11, 0.65, 0.24	0.48 (M), 0.52 (F)

missing values after removing those students who drop out. Due to the number of samples already available for this dataset, we discard samples with missing values.

4.7.4. DAP

The dataset of academic performance evolution for engineering students (DAP) (Delahoz-Dominguez et al., 2020) can be used to predict the global performance of engineering students in Colombian universities based on test scores achieved in the final year of high school and a diverse set of demographic information. Usually, this dataset is used for regression tasks, but we recode the global score into four categories. As most variables in this dataset have the explicit option of "Missing" information, we recoded true missing values into this category.

4.7.5. ELP

The early learning predictors (ELP) dataset[5] was presented as part of a challenge on the data science platform Zindi and contains information about children in South Africa (mostly of the age of five). The dataset includes demographic information as well as physical, social, emotional, and cognitive assessments, along with details about the school environment. The prediction goal is to classify each child into one of three categories: "falling far behind", "falling behind", and "on track". These categories are determined from multiple assessments, so the dataset captures more than just academic performance. Some features were dropped due to excessive missing values. For the remaining features, missing values were recoded as a separate category, since in this context, missing information may itself carry meaningful information. As labels are not available for the challenge's test set, we use only the training dataset.

In total, the datasets vary significantly across many dimensions: the ratio of features to samples, the age group of the students, the region of the world, the distribution of the target variable, and the prediction goals. In particular, the diversity of feature and sample sizes is important because structure learning methods are known to be affected by these data characteristics.

5. Results

We now discuss the evaluation results. Given that we received results for 259 models per dataset, we cannot show all results and selected results based on performance[6].

[5] https://zindi.africa/competitions/datadrive2030-early-learning-predictors-challenge/data

[6] All results can be found online.

5.1. Predictive Performance

Table 2 presents the results on predictive performance. With respect to **RQ1**, we can say that the models trained using features selected via CFS usually perform on-par with the baseline. The only negative exceptions are the DAP dataset and the xAPI dataset to an extent. With respect to the comparison to the best non-causal filter method (again, the CFS methods formally belong to the filter FS family), we observe that the best CFS method leads to ML models that sometimes perform better and sometimes worse, with no clear winner apart from the DAP dataset, with which the causal methods struggle indicating that the DAG is tricky to learn. The DAP dataset may be particularly tricky to learn because it has a relatively small feature-sample ratio; then again, OULAD has an even smaller one. Another explanation of this pattern may be that we recoded the target variable by binning it. Learning the relationship of other variables with this newly recoded target may be particularly difficult in this setting.

More generally, it can be said that there is no general disadvantage when using CFS instead of non-causal filter methods. Moreover, the percentage of the selected features from the whole dataset is typically lower when using CFS methods, which has advantages with regard to efficiency and interpretability.

With respect to the comparison to the best overall non-causal FS method, we observe that the best method is typically a wrapper method and outperforms both the best non-causal filter method and the best CFS method. However, for OULAD, e.g., the F1-value achieved with the feature subsets derived via CFS is still generally close. It is interesting to observe that the best method is usually recursive feature elimination. The concrete ML method that works best with this FS method varies and seems to be quite independent of the ML method used to train the model. For the CFS methods, there is no such clear best method, although HC-MB appears most often, followed by IAMB. For the best filter method, there is also no clear best method. Again, what is very noticeable and interesting is that the best CFS method typically returns a smaller, sometimes much smaller, feature subset than the best non-causal CFS method. For the SAP dataset, the best CFS method returned a subset bigger than the subset returned by the best non-causal FS method for four ML methods; for the OULAD dataset and the DAP dataset, that happens once each. For all other datasets and ML methods, the best CFS method returns a smaller subset. For the DAP and ELP datasets, HC-MB returns about 1% of features only. These are clearly very relevant, as the results for the best non-causal FS methods are better yet not that much better, while using many, many more features. This observation shows that the CFS methods are capable of returning truly relevant features that are also potentially well-suited for interventions and that they lead to more efficient ML models, as we expected. Generally, **RQ2** has to be answered by stating that the best non-causal filter FS method performs about as well as the best CFS method and that the best overall non-causal FS method outperforms or is at least generally equal to the best CFS method, although the differences vary by dataset. This variance makes sense as DAG learning also varies significantly by dataset. There is no clearly best CFS method but HC typically performs well.

5.2. Fairness

Table 3 shows the results for **RQ3** and **RQ4** when considering the DPR. The results for EOR are available in our online repository only because they show the same patterns. We only show results for the first three datasets because the last two (DAP, ELP) never return a DPR < 0.8 for any constellation (using or not using CFS and fair pre-processing), indicating that fairness

Table 2: A comparison of the performance (F1) of each ML method without feature selection, with the best non-causal filter feature selection, the best non-causal feature selection, and causal feature selection method. The percentage of features selected by the respective FS method is also presented. The overall best result for a dataset is bold. A result that significantly ($p < 0.05$) deviates from the baseline using the whole dataset (positively or negatively) is printed in italics. If two methods are named, their results are exactly identical (beyond the second decimal point).

Dataset	ML Method	Whole Data F1	Best FilterFS F1	% Features Selected	Best FS F1	% Features Selected	Best CFS F1	% Features Selected
	kNN	0.63	MI – 0.63	0.80	RFE_LR – *0.70*	0.40	IAMB, LRH – 0.63	0.21
	SVM	0.63	SelectF – 0.63	0.20	RFE_LR – *0.74*	0.40	HITON – *0.68*	0.10
	LR	0.77	SelectF – 0.76	0.40	RFE_XGB – 0.77	0.60	HC-MB – 0.75	0.26
xAPI	RF	0.80	MI – 0.79	0.80	RFE_XGB – **0.81**	0.60	HC-MB – 0.77	0.26
	XGBoost	0.78	MI – 0.78	0.80	RFE_RF – 0.78	0.80	HC-MB – *0.72*	0.26
	CatBoost	0.80	SelectF – 0.79	0.80	RFE_XGB – 0.80	0.80	HC-MB – *0.75*	0.26
	NN	0.74	MI – 0.75	0.80	RFE_RF/MI – 0.75	0.80	HC-MB – 0.75	0.26
	kNN	0.47	MI – *0.58*	0.20	RFE_LR – *0.65*	0.20	HC-MB – 0.48	0.52
	SVM	0.45	MI – *0.57*	0.20	RFE_LR – *0.60*	0.20	HC-MB – 0.47	0.52
	LR	0.53	SelectF – 0.56	0.20	RFE_XGB – *0.58*	0.40	HC-MB – 0.56	0.52
SAP	RF	0.50	SelectF – *0.58*	0.20	FS_XGB – **0.67**	0.36	IAMB, LRH – *0.60*	0.52
	XGBoost	0.58	SelectF – 0.61	0.80	L1 – *0.62*	0.72	HC-MB – *0.62*	0.52
	CatBoost	0.60	MI – 0.57	0.40	RFE_LR – 0.62	0.20	HC-MB – 0.62	0.52
	NN	0.52	SelectF – 0.54	0.80	RFE_LR – *0.58*	0.20	HC-MB – 0.53	0.52
	kNN	0.72	MI – 0.72	0.80	RFE_XGB – *0.76*	0.20	IAMB, LRH – 0.73	0.63
	SVM	0.73	MI – 0.73	0.80	RFE_XGB – 0.73	0.80	IAMB, LRH – 0.73	0.63
	LR	0.77	MI – 0.76	0.80	RFE_XGB – 0.77	0.40	IAMB, LRH – 0.77	0.63
OULAD	RF	0.79	MI – 0.79	0.80	RFE_RF – 0.79	0.60	IAMB, LRH – 0.76	0.63
	XGBoost	0.78	MI – 0.78	0.80	RFE_XGB – 0.79	0.80	IAMB, LRH – 0.78	0.63
	CatBoost	0.79	MI – 0.79	0.80	RFE_XGB – **0.79**	0.80	IAMB, LRH – **0.79**	0.63
	NN	0.70	SelectF – *0.77*	0.80	SFS_RF – *0.78*	0.76	IAMB, LRH – *0.79*	0.63
	kNN	0.48	SelectF – 0.48	0.80	RFE_XGB – 0.48	0.40	HC-MB – *0.41*	0.01
	SVM	0.44	SelectF – 0.46	0.20	Boruta – *0.47*	0.13	HC-MB – 0.43	0.01
	LR	0.34	MI – 0.34	0.80	L1 – 0.34	0.95	HC-MB – *0.43*	0.01
DAP	RF	0.51	MI – 0.51	0.40	MI – 0.51	0.40	IAMB – 0.47	0.60
	XGBoost	0.50	MI – 0.51	0.60	RFE_XGB – 0.51	0.80	IAMB – *0.45*	0.60
	CatBoost	0.51	MI – 0.51	0.80	L1 – **0.51**	0.95	IAMB – 0.46	0.60
	NN	0.49	MI – 0.48	0.80	MI – 0.48	0.80	HC-MB – *0.35*	0.01
	kNN	0.34	SelectF – *0.52*	0.60	RFE_LR – *0.57*	0.40	HC-MB – *0.50*	0.01
	SVM	0.26	All – 0.26	All	RFE_LR – *0.51*	0.40	HC-MB – *0.33*	0.01
	LR	0.36	SelectF – *0.42*	0.40	RFE_LR – *0.59*	0.60	HC-MB – *0.45*	0.01
ELP	RF	0.56	SelectF – 0.56	0.20	RFE_LR – 0.57	0.40	HC-MB – *0.50*	0.01
	XGBoost	0.58	MI – 0.58	0.80	L1 – 0.58	0.8	HC-MB – *0.51*	0.01
	CatBoost	0.57	SelectF – 0.59	0.80	RFE_LR – **0.59**	0.40	HC-MB – *0.51*	0.01
	NN	0.43	SelectF – *0.53*	0.80	RFE_LR – *0.58*	0.60	IAMB – 0.42	0.28

Table 3: A comparison of the predictive performance (F1) and fairness in terms of the DPR of the baseline using all features with the best performing fair CFS method, the fairest fair CFS method, the best fair other pre-processing method, and the fairest fair other pre-processing method for each dataset and ML method. The baseline is used as a reference; the difference from the baseline value is computed for all other results. A plus (+) indicates a positive change, a minus (-) a negative change. If there is no change up to the second decimal place, the value is simply zero. A value in italics represents a significant ($p < 0.05$) deviation from the baseline. The first number refers to the F1 value, the second to the DPR.

Dataset	ML Method	Whole Data	Best Fair CFS	Fairest Fair CFS	Best Fair PP	Fairest P.P.
xAPI	kNN	0.63/0.59	-0.01/+*0.18* (CaDe)	-0.01/+*0.18* (CaDe)	+0.02/+*0.15* (CR)	+0.02/+*0.15* (CR)
	SVM	0.63/0.65	*-0.07*/+*0.08* (CaDe)	*-0.07*/+*0.08* (CaDe)	0.00/+0.02 (CR)	-0.02/+*0.08* (DR)
	LR	0.77/0.59	-0.04/+*0.09* (CaDe)	-0.04/+*0.09* (CaDe)	-0.01/+0.02 (DR)	*-0.11*/+*0.13* (FA-FS-LR)
	RF	0.80/0.58	-0.03/+0.02 (CaDe)	*-0.44*/+*0.10* (FairCFS)	0.00/-0.02 (CR)	*-0.16*/+*0.11* (FA-FS-LR)
	XGBoost	0.78/0.62	*-0.04*/0.00 (CaDe)	*-0.22*/+*0.06* (ExDesc)	+0.01/-0.01 (CR)	*-0.13*/+*0.06* (FA-FS-LR)
	CatBoost	0.80/0.57	*-0.04*/0.01 (CaDe)	*-0.33*/+*0.16* (ExDesc)	0.00/+*0.05* (CR)	0.00/+*0.08* (FA-FS-XGBoost)
	NN	0.74/0.61	-0.02/+*0.13* (CaDe)	-0.02/+*0.13* (CaDe)	-0.01/0.00 (DR)	*-0.08*/+*0.18* (CR)
SAP	kNN	0.47/0.00	0.00/+*0.74* (MBExDesc)	0.00/+*0.74* (MBExDesc)	+*0.14*/+*0.53* (FA-FS-RF)	+0.04/+*0.71* (FA-FS-LR)
	SVM	0.45/0.00	+0.03/+*0.58* (CaDe)	0.00/+*0.66* (ExDesc)	+*0.05*/+*0.70* (FA-FS-RF)	0.00/+*0.71* (FA-FS-LR)
	LR	0.53/0.49	+0.03/0.00 (CaDe)	+0.03/+0.01 (MBExDesc)	+0.02/+0.06 (FA-FS-RF)	*-0.08*/+*0.18* (FA-FS-LR)
	RF	0.50/0.00	+*0.13*/+*0.48* (MBExDesc)	+*0.10*/+*0.61* (ExDesc)	+*0.16*/+*0.42* (FA-FS-RF)	-0.09/+*0.71* (DR)
	XGBoost	0.58/0.00	+*0.06*/+*0.46* (MBExDesc)	+0.03/+*0.57* (ExDesc)	+0.03/+*0.50* (FA-FS-RF)	*-0.14*/+*0.66* (DR)
	CatBoost	0.60/0.00	+0.02/+*0.54* (CaDe)	+0.02/+*0.54* (CaDe)	0.00/+*0.54* (FA-FS-RF)	*-0.09*/+*0.62* (FA-FS-LR)
	NN	0.52/0.00	+0.01/+*0.53* (CaDe)	0.00/+*0.58* (ExDesc)	+0.02/+*0.55* (CR)	-0.05/+*0.65* (FA-FS-LR)
OULAD	kNN	0.72/0.83	0.00/+*0.11* (CaDe)	0.00/+*0.11* (CaDe)	+0.02/+0.01 (CR)	+0.01/+*0.15* (DR)
	SVM	0.73/0.71	+0.02/-0.01 (CaDe)	*-0.14*/+*0.19* (ExDesc)	+0.02/+*0.24* (DR)	+0.02/+*0.24* (DR)
	LR	0.77/0.77	0.00/0.00 (CaDe)	*-0.21*/+*0.08* (ExDesc)	0.00/+0.02 (DR)	0.00/+*0.19*(FA-FS-LR)
	RF	0.79/0.90	-0.01/*-0.05* (CaDe)	*-0.10*/0.00 (ExDesc)	0.00/-0.02 (CR)	*-0.03*/+*0.08* (FA-FS-LR)
	XGBoost	0.78/0.89	+*0.03*/*-0.18* (CaDe)	*-0.20*/0.00 (ExDesc)	+0.01/+0.02 (CR)	-0.02/+*0.08* (FA-FS-LR)
	CatBoost	0.79/0.89	*-0.04*/*-0.09* (CaDe)	*-0.21*/*-0.02* (ExDEsc)	0.00/+0.01 (CR)	+0.02/+*0.08* (FA-FS-LR)
	NN	0.70/0.85	+0.04/-0.05 (CaDe)	*-0.13*/0.00 (ExDesc)	+0.06/-0.05 (DR)	+0.06/+*0.07* (CR)

constraints are unnecessary and that they make little difference. The table shows the results of the baseline using all features with respect to predictive performance and each fairness measure. The best performing and fairest fair CFS methods and other fair pre-preocessing methods are compared to it, with positive values indicating a positive and negative values indicating a negative change. The results of the best method according to the chosen metric are displayed; if two methods are identical according to that metric (e.g., DPR), the results of the method are displayed that scored higher on the other metric looked at (e.g., F1).

As expected, there is usually a trade-off between performance and fairness. A major exception (except for the fairest pre-processing methods) is the SAP dataset, where performance also increases. The fair CFS methods and fair pre-processing methods show similar patterns, and it is not always clear which performs better. Only for some combinations of the OULAD dataset can it be argued that fair pre-processing methods perform better, though the dataset already shows little bias. Within the fair CFS methods, *ExDec* and *CaDe* seem to perform best. All in all, we may answer **RQ3** by stating that there is no large difference between causal and non-causal fair pre-processing methods. **RQ4** can be answered by saying that a trade-off certainly exists, but that its exact magnitude depends on the specific scenario.

As a consequence, we may cautiously conclude the following: causal structure learning methods uncover causal structures that are truly relevant, as evidenced by the mostly competitive performance of CFS on downstream tasks—at least compared to other filter methods. However, their performance varies by dataset, likely reflecting that causal structure learning is tougher in some settings. CFS typically returns far fewer features, leading to greater efficiency.

5.3. Stability of MBs

A concern one may voice regarding the eventual use of the causal model beyond increasing predictive performance and mitigating algorithmic bias (e.g., for constructing interventions) is whether the MBs learned are identical across all folds, indicating that the causal information encoded is relatively stable. If results are relatively stable, this would further evoke trust, because the methods are not reactive to small changes in the data. Therefore, we examined the MBs returned by HC-MB (the method leading to the best overall results) and compared the variables in the MBs across folds for each dataset. HC-MB returned the same MB across all folds for OULAD, DAP, and ELP. For xAPI, the same MB was returned four times, but one fold led to a slightly different MB, which included an additional variable. For SAP, three distinct MBs were returned across the five folds; however, two of them (equivalent to four folds) only differed by one variable. The other MB was also vastly similar, with two differences compared to the other two MBs. This result indicates a relatively large stability. At the same time, as we are only considering the MB here and not whether this also translates to the whole DAG—theoretically, we could receive the same MB even when the DAG differs—we should still treat this conclusion carefully.

6. Conclusion, Limitations and Future Work

Causal methods offer many theoretical benefits over non-causal methods that are especially important in EDM. Yet, real-world applications of causal methods are scarce because they require causal graphs, which are challenging to construct. This study demonstrates that causal methods retain value even when DAGs are not learned perfectly accurately. For predictive performance, we saw that causal methods typically at least achieved a performance equivalent to the baseline and the other filter methods; given the often drastic reduction of the feature space, this finding shows that causal methods are of interest for FS, although they often lead to less accurate models compared to wrapper FS methods. For algorithmic bias, fair causal methods often offer a reasonable trade-off between bias and predictive performance, with the methods not clearly worse than other fair pre-processing methods. These results indicate that the constructed graphs can contain valuable insights for, e.g., interventions—even if they are not perfectly correct. Generally, our findings tentatively support the call for causal methods in EDM. Particularly, because we are also interested in insights for interventions, causal approaches are desirable beyond the mere performance.

Of course, we should still be careful about any additional insights we try to gain from the graphs. We encourage practitioners not to blindly trust the graphs, but rather to embed more background knowledge, potentially following a circular improvement scheme that restricts or fixes certain edges and relearns the structure based on this. This procedure could not only provide better insights but also drastically improve predictive performance and reduce algorithmic bias by leading to more truly causal graphs. Given the very limited background knowledge we used here, our results even represent something like a worst-case outcome.

In addition, it seems to be the case that the causal methods had more trouble with specific datasets (notably, DAP) than with others. The only working hypothesis we have to explain this pattern is that our recoding of the target variable may be relevant. Generally, structure learning methods are known to struggle with non-linear and complex data, which certainly means that not every dataset is equally well-suited for employing causal methods. Future research should

further examine which dataset characteristics make it harder for causal methods to learn the relevant structures.

Moreover, we do not know exactly how close or how far away our DAGs are from the (unavailable) ground truth. Consequently, future work should investigate (then again using synthetic data) how changing, removing, and adding edges in an increasingly incorrect manner affects the performance on subsequent downstream tasks. It should be noted that it is unclear what patterns of errors emerge using real data (e.g., are edges incorrectly oriented or removed); for this reason, we used real data directly. However, we believe that this is a very interesting future research endeavor.

Another limitation of our work, as one might claim, is that the number of datasets we used remains limited. While true, we tried to use diverse datasets, and even now, reporting all the results is already challenging. Still, other kinds of predictive tasks should be looked at in the future (e.g., regression tasks, clickstream data). With regard to the methods, it could be criticized that we could have compared different or more causal structure learning methods. We selected methods we deemed well-fitting, but there are certainly others, and given that few research works employed these methods on real-life data, it is not yet clear which methods might be particularly promising. We see our paper as a contribution to judging the methods accordingly. The same can be said for the fair causal methods, of which there are fewer to date. We encourage future research in the direction of learning DAGs more accurately and developing fair methods.

Finally, our study only demonstrates the general usability of causal methods in EDM; we have not yet gone further and looked at the extraction of relevant information (e.g., for interventions) from the learned graphs.

REFERENCES

AMRIEH, E. A., HAMTINI, T., ALJARAH, I., ET AL. 2016. Mining educational data to predict student's academic performance using ensemble methods. *International journal of database theory and application 9,* 8, 119–136.

BATOOL, S., RASHID, J., NISAR, M. W., KIM, J., KWON, H.-Y., AND HUSSAIN, A. 2023. Educational data mining to predict students' academic performance: A survey study. *Education and Information Technologies 28,* 1, 905–971.

BELITZ, C., JIANG, L., AND BOSCH, N. 2021. Automating procedurally fair feature selection in machine learning. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 379–389.

BELLAMY, R. K., DEY, K., HIND, M., HOFFMAN, S. C., HOUDE, S., KANNAN, K., LOHIA, P., MARTINO, J., MEHTA, S., MOJSILOVIĆ, A., ET AL. 2019. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development 63,* 4/5, 4–1.

BERTOLINI, R., FINCH, S. J., AND NEHM, R. H. 2021. Enhancing data pipelines for forecasting student performance: integrating feature selection with cross-validation. *International Journal of Educational Technology in Higher Education 18,* 1, 44.

BIRD, S., DUDÍK, M., EDGAR, R., HORN, B., LUTZ, R., MILAN, V., SAMEKI, M., WALLACH, H., AND WALKER, K. 2020. Fairlearn: A toolkit for assessing and improving fairness in ai.

CHANDRASHEKAR, G. AND SAHIN, F. 2014. A survey on feature selection methods. *Computers & electrical engineering 40,* 1, 16–28.

COHAUSZ, L. 2025. Why the future of aied is causal: Arguments for creating a tradition based on causal thinking. In *International Conference on Artificial Intelligence in Education*. Springer, 17–31.

COHAUSZ, L., KAPPENBERGER, J., AND STUCKENSCHMIDT, H. 2025. Causal graphs and fairness in machine learning: addressing practical challenges in causal fairness evaluation. *Journal of Artificial Intelligence Research 84*.

COHAUSZ, L., TSCHALZEV, A., BARTELT, C., AND STUCKENSCHMIDT, H. 2023. Investigating the importance of demographic features for edm-predictions. *International Educational Data Mining Society*.

DELAHOZ-DOMINGUEZ, E., ZULUAGA, R., AND FONTALVO-HERRERA, T. 2020. Dataset of academic performance evolution for engineering students. *Data in brief 30*, 105537.

DWORK, C., HARDT, M., PITASSI, T., REINGOLD, O., AND ZEMEL, R. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, Cambridge Massachusetts, 214–226.

ERICKSON, N., PURUCKER, L., TSCHALZEV, A., HOLZMÜLLER, D., DESAI, P. M., SALINAS, D., AND HUTTER, F. 2025. Tabarena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*.

GALHOTRA, S., SHANMUGAM, K., SATTIGERI, P., AND VARSHNEY, K. R. 2022. Causal feature selection for algorithmic fairness. In *Proceedings of the 2022 International Conference on Management of Data*. 276–285.

GUO, X., YU, K., CAO, F., LI, P., AND WANG, H. 2022. Error-aware markov blanket learning for causal feature selection. *Information Sciences 589*, 849–877.

GUYON, I., ALIFERIS, C., ET AL. 2007. Causal feature selection. In *Computational methods of feature selection*. Chapman and Hall/CRC, 79–102.

HARDT, M., PRICE, E., AND SREBRO, N. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems 29*.

HUSSAIN, S. 2018. Student Academics Performance. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50W30.

JOVIĆ, A., BRKIĆ, K., AND BOGUNOVIĆ, N. 2015. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee, 1200–1205.

KITSON, N. K., CONSTANTINOU, A. C., GUO, Z., LIU, Y., AND CHOBTHAM, K. 2023. A survey of bayesian network structure learning. *Artificial Intelligence Review 56,* 8, 8721–8814.

KURSA, M. B., JANKOWSKI, A., AND RUDNICKI, W. R. 2010. Boruta–a system for feature selection. *Fundamenta informaticae 101,* 4, 271–285.

KUSNER, M. J., LOFTUS, J., RUSSELL, C., AND SILVA, R. 2017. Counterfactual fairness. *Advances in neural information processing systems 30*.

KUZILEK, J., HLOSTA, M., AND ZDRAHAL, Z. 2017. Open university learning analytics dataset. *Scientific data 4,* 1, 1–8.

LING, Z., LI, Y., ZHANG, Y., YU, K., ZHOU, P., LI, B., AND WU, X. 2022. A light causal feature selection approach to high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering 35,* 8, 7639–7650.

LING, Z., XU, E., ZHOU, P., DU, L., YU, K., AND WU, X. 2024. Fair feature selection: A causal perspective. *ACM Transactions on Knowledge Discovery from Data 18,* 7, 1–23.

LIU, X. AND LIU, X. 2016. Swamping and masking in markov boundary discovery. *Machine Learning 104,* 1, 25–54.

MIHAESCU, M. C. AND POPESCU, P. S. 2021. Review on publicly available datasets for educational data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 11,* 3, e1403.

PEARL, J. 2009. *Causality*. Cambridge university press.

RAMASWAMI, M. AND BHASKARAN, R. 2009. A study on feature selection techniques in educational data mining. *arXiv preprint arXiv:0912.3924*.

SAARELA, M. 2024. On the relation of causality-versus correlation-based feature selection on model fairness. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. 56–64.

SCHMIDT, M. 2005. Least squares optimization with l1-norm regularization. *CS542B Project Report 504,* 2005, 195–221.

WAH, Y. B., IBRAHIM, N., HAMID, H. A., ABDUL-RAHMAN, S., AND FONG, S. 2018. Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science & Technology 26,* 1.

WEIDLICH, J., HICKS, B., AND DRACHSLER, H. 2024. Causal reasoning with causal graphs in educational technology research. *Educational technology research and development 72,* 5, 2499–2517.

YU, K., GUO, X., LIU, L., LI, J., WANG, H., LING, Z., AND WU, X. 2020. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys (CSUR) 53,* 5, 1–36.

YU, K., LIU, L., AND LI, J. 2021. A unified view of causal and non-causal feature selection. *ACM Transactions on Knowledge Discovery from Data (TKDD) 15,* 4, 1–46.

YU, K., YANG, Y., AND DING, W. 2022. Causal feature selection with missing data. *ACM Transactions on Knowledge Discovery from Data (TKDD) 16,* 4, 1–24.

ZAFFAR, M., HASHMANI, M. A., SAVITA, K., AND KHAN, S. A. 2021. A review on feature selection methods for improving the performance of classification in educational data mining. *International Journal of Information Technology and Management 20,* 1-2, 110–131.