


# Bias in Machine Learning: A Literature Review

Konstantinos Mavrogiorgos , Athanasios Kiourtis , Argyro Mavrogiorgou \* , Andreas Menychtas   
and Dimosthenis Kyriazis 

Department of Digital Systems, University of Piraeus, 185 34 Piraeus, Greece; komav@unipi.gr (K.M.);  
kiourtis@unipi.gr (A.K.); amenychtas@unipi.gr (A.M.); dimos@unipi.gr (D.K.)

\* Correspondence: margy@unipi.gr

**Abstract:** Bias could be defined as the tendency to be in favor or against a person or a group, thus promoting unfairness. In computer science, bias is called algorithmic or artificial intelligence (i.e., AI) and can be described as the tendency to showcase recurrent errors in a computer system, which result in “unfair” outcomes. Bias in the “outside world” and algorithmic bias are interconnected since many types of algorithmic bias originate from external factors. The enormous variety of different types of AI biases that have been identified in diverse domains highlights the need for classifying the said types of AI bias and providing a detailed overview of ways to identify and mitigate them. The different types of algorithmic bias that exist could be divided into categories based on the origin of the bias, since bias can occur during the different stages of the Machine Learning (i.e., ML) lifecycle. This manuscript is a literature study that provides a detailed survey regarding the different categories of bias and the corresponding approaches that have been proposed to identify and mitigate them. This study not only provides ready-to-use algorithms for identifying and mitigating bias, but also enhances the empirical knowledge of ML engineers to identify bias based on the similarity that their use cases have to other approaches that are presented in this manuscript. Based on the findings of this study, it is observed that some types of AI bias are better covered in the literature, both in terms of identification and mitigation, whilst others need to be studied more. The overall contribution of this research work is to provide a useful guideline for the identification and mitigation of bias that can be utilized by ML engineers and everyone who is interested in developing, evaluating and/or utilizing ML models.



**Citation:** Mavrogiorgos, K.; Kiourtis, A.; Mavrogiorgou, A.; Menychtas, A.; Kyriazis, D. Bias in Machine Learning: A Literature Review. *Appl. Sci.* **2024**, *14*, 8860. <https://doi.org/10.3390/app14198860>

Academic Editors: Rui Araújo and Lykourgos Magafas

Received: 27 August 2024

Revised: 21 September 2024

Accepted: 23 September 2024

Published: 2 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** bias; algorithms; machine learning; artificial intelligence; literature review

## 1. Introduction

Bias is the tendency to promote prejudiced results due to erroneous assumptions. In the context of ML and AI in general, this could be caused due to erroneous data on which an ML model has been trained, or due to other factors that will be further analyzed later in this manuscript. Algorithmic or not [1], bias is not a recent discovery and has been recognized by the research community for many decades. Looking back in history, it is quite clear that bias not only exists but also has played—and still plays—a crucial negative role in the evolution of humanity since the very early stages of its existence. Regardless of the era that one lives in, the concept of bias remains the lack of internal validity or even false assessment between an exposure and an effect that is expected to have a certain outcome that does not equal the real value [2]. Bias from the “outside” world can infiltrate AI systems through, for instance, the data that have been used for training the underlying ML models, thus affecting the results of the said systems. In essence, bias finds its way into AI systems and, given the fact that the adaptation of those systems is increasing exponentially [3], it is of vital importance to identify it and try to mitigate it, by utilizing a plethora of tools, algorithms and approaches that people have in their disposal. It is no surprise that in a very recent survey [4], bias is, alongside data privacy, one of the key aspects about which people are concerned when using AI tools.

There have been several examples of AI tools, which have even been exploited in a production environment, that have produced erroneous results and predictions due to bias in the input data and/or the training process of the corresponding ML models. In deeper detail, an indicative example of such tools is Large Language Models (LLMs) like ChatGPT, which are known for inheriting bias from the training data, since the data themselves are not bias-free [5,6]. This is a well-known issue of LLMs for which extensive research is being carried out to minimize such bias. Bias has occurred repeatedly in computer vision (CV) algorithms and applications as well. For example, in 2020 it was identified that the algorithm used by Twitter (now X) for cropping images favored light-skinned over dark-skinned individuals, whilst it also favored cropping female bodies instead of their faces, thus showcasing signs of both racial and gender bias [7]. This is not the only example of such an algorithm since both algorithms developed by Google and Apple for editing images have also showcased similar behaviors [8]. In those cases, the training data were also the culprit for the biased results generated by the algorithms. In another case, which sparked a nation-wide debate in the United States, a tool called COMPAS that was used as a Recidivism Prediction Instrument (RPI), was found to be biased in terms of race. The goal of this tool was to predict the probability of a criminal defendant reoffending at some time into the future [9]. It was discovered that this tool was biased against black defendants, and this was not only due to the training data but also due to the way that the underlying algorithm operates [10]. In the case of COMPAS, the nature of the algorithm relied on population effects, which were also wrong because the training data were biased. The law cares about individual cases and not population effects [11]; thus, the nature of the ML algorithm utilized in COMPAS was also responsible for the biased results that it produced.

The above highlights the vital importance of mitigating any form of bias that might occur in AI tools in different domains since the presence of bias will inevitably lead to wrongful decision making. The vast number of different algorithmic biases that exist have led to their classification, based on specific characteristics, in order to be easier to study them and propose appropriate ways to deal with them. Such a classification that covers all the different kinds of AI bias is based on the origin of bias. The bias that can be found in an AI system (i.e., algorithmic bias) can generally be derived from three sources. The first source is, obviously the data that the system utilizes, since if, for instance, an ML model is trained on biased data, it will produce biased results [12]. The second source is the design of the algorithm and the way that it operates. For instance, if the algorithm utilized does not fit the given problem, it may lead to biased outputs. In another example, the objective functions used in optimization algorithms may inject bias to maximize utility or minimize costs [13]. The third source of algorithmic bias is the human bias. Of course, bias found in the data can also be due to human bias. However, in the context of this paper, the reference to human bias as a “high-level” category of bias refers to the bias that originates from the ones that develop and/or evaluate an ML model [14]. An indicative example of such a bias is the confirmation bias, where ML engineers unconsciously process data in a way that affirms an initial hypothesis or a specific goal [15].

In the literature, there have been several attempts to identify the different forms of algorithmic bias that exist in order to guide the corresponding implementations towards bias-free AI tools. To this end, this manuscript summarizes the different types of algorithmic bias that have been identified in the literature by classifying them into one of the following three categories, namely data bias, algorithm bias and engineer bias, which are also essential parts of the ML lifecycle. Alongside the type of algorithmic bias, the corresponding methods that have been proposed in the literature are also mentioned, thus serving as a handbook for everyone who is interested in developing, evaluating and/or using an AI model and aims to identify and mitigate bias in such a model. What is more, this manuscript identifies that most of the literature focuses mostly on a specific type of bias, namely the data bias, whilst it completely omits other aspects of the ML lifecycle that can introduce bias. As a result, it provides a future direction with regard to other aspects of the ML lifecycle that

should be better investigated and clarified in terms of the way that they can introduce bias and in what way this bias can be addressed.

The rest of the manuscript is structured as follows. Section 2 describes the methodology that was followed in order to perform the comprehensive literature review that is presented in this manuscript. Section 3 reports the types of algorithmic bias that originate from the data themselves and the diverse methods that have been proposed in the literature in order to mitigate them. More specifically, it analyzes general approaches that exist in the literature regarding the mitigation of bias in data and then dives into specific types of data bias, namely cognitive, selection and reporting, and presents the different methods and techniques that have been presented in the literature. Section 4 analyzes the set of algorithmic biases that are caused by the algorithms, as well as the corresponding approaches to deal with them. In deeper detail, this section analyzes how estimators, optimizers and different regularization methods can introduce bias and presents several comparative studies per different domains that aim to find the approach that achieves the least possible bias. In this section, specific bias-mitigation algorithms are also presented and analyzed. Section 5 describes the group of algorithmic biases that originate from the model builders and model evaluators and studies the analogous countermeasures that are proposed in the literature. Section 6 discusses the proposed approaches for the mitigation of algorithmic bias presented in the above-mentioned sections and potential unclear points that should be further researched. It also compares this literature review with other similar reviews and highlights the added value of this manuscript. Lastly, Section 7 summarizes the findings of this manuscript and provides future research directions with regard to bias identification and mitigation in ML. The sections of this manuscript are also depicted in Table 1.

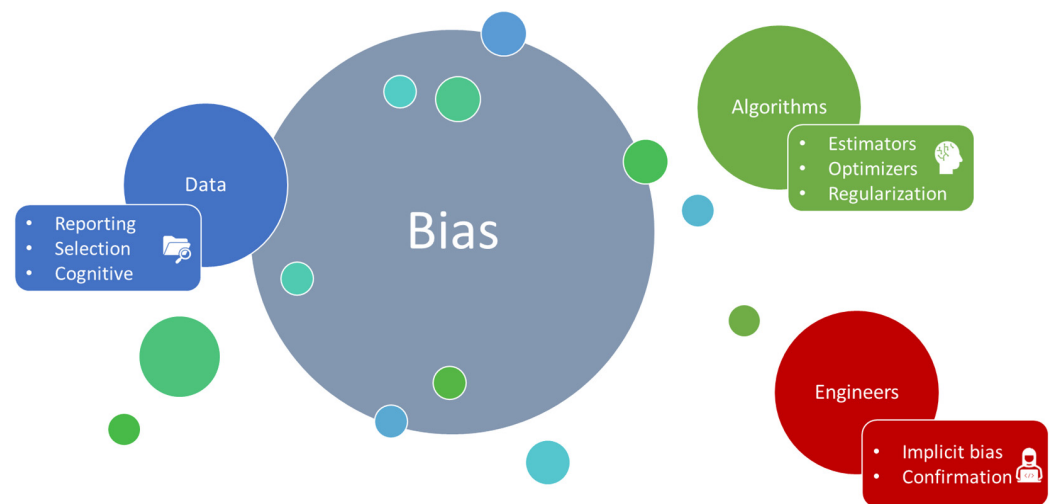
**Table 1.** Table of the contents of this manuscript.

Section	Description
Section 1	Introduction to the scope of this manuscript and the overall work conducted
Section 2	Methodology followed to carry out the comprehensive literature review
Section 3	Analysis of approaches for identifying and mitigating data bias
Section 3.1	Analysis of approaches for identifying cognitive bias in data
Section 3.2	Analysis of approaches for identifying selection bias in data
Section 3.3	Analysis of approaches for identifying reporting bias in data
Section 3.4	Summarization of the most common approaches for mitigating data bias based on the findings of Sections 3.1–3.3
Section 4	Analysis of approaches for identifying and mitigating algorithm bias
Section 4.1	Research regarding the different estimators that have been utilized in the literature, in diverse domains, and the way that they may introduce bias
Section 4.2	Research regarding the different optimizers that have been utilized in the literature, in diverse domains, and the way that they may introduce bias
Section 4.3	Research regarding the different regularization methods that have been utilized in the literature, in diverse domains, and the way that they may introduce bias
Section 4.4	Summarization of the most common approaches for mitigating algorithm bias based on the findings of Sections 4.1–4.3
Section 5	Analysis of approaches for identifying and mitigating engineer bias
Section 6	Discussion of the findings of this manuscript and comparison to other literature reviews about bias
Section 7	Summarization of the findings of this manuscript and provision of future research directions with regard to bias in ML

## 2. Materials and Methods

As mentioned in the Introduction part, this manuscript aims to perform a comprehensive literature review regarding bias in ML, investigating the corresponding methods and approaches that have been proposed in order to identify and/or mitigate bias in the whole ML lifecycle. More specifically, in the context of this manuscript, the ML lifecycle consists of three (3) discrete stages for each there exist specific types of bias: (i) bias that can

originate from the data themselves (i.e., data bias), (ii) bias deriving from the ML models that are utilized (i.e., algorithm bias) or, (iii) bias by the ML engineers that develop and/or evaluate the produced ML models (i.e., engineer bias). Having divided the ML lifecycle into those categories, the types of bias that can occur in each category were identified, which on the one hand allowed the definition of the said biases, and on the other hand aided the identification of the corresponding approaches in the literature that manage to tackle those biases. An overview of the types of bias that can occur in each of the three (3) categories can be shown in Figure 1.



**Figure 1.** Types of bias per ML lifecycle category.

When the different types of bias were identified, four (4) of the most widely known publications databases were selected, namely ACM Digital Library, IEEE Xplore, Scopus and Science Direct. For each database, the corresponding search query was generated. The search for relevant studies was based on the presence of specific search terms in the title, abstract or keywords of the said studies. For the data bias, the search terms consisted of the name of the corresponding bias (e.g., reporting bias), the phrase “machine learning”, since this manuscript is focused on ML approaches that aim to address the issue of bias presence in data, and words like “addressing”, “mitigating”, “identifying” to further limit the search to studies that actually propose a way to identify and/or mitigate data bias. Regarding algorithm bias, more specifically estimator bias, another set of search terms was used. In the estimator bias, only studies that compare different algorithms and also take into consideration the aspect of bias were needed. As a result, words like “bias”, “comparative study”, and “review” were used as search terms. Similarly, regarding the optimizers’ bias, search terms like “bias”, “optimizer”, “hyperparameter”, and “neural network” were used. The “hyperparameter” search term was used to only limit the search to studies that do not refer to optimization algorithms in general, but to them as a hyperparameter of neural networks. As for the bias caused by regularization techniques, a similar approach was followed, consisting of the search terms “regularization”, “machine learning” and the names of widely used regularization techniques. Lastly, regarding the engineer bias, there exist no studies that particularly address this bias from the point of view of ML engineers, rather than the point of view of researchers in general. As a result, for this specific type of bias, and its subtypes, no specific search queries were performed, and a more theoretical framework is analyzed in the corresponding section of this manuscript. A complete list of the search queries that were performed per type of bias and per publications’ database is shown in Table 2.

**Table 2.** List of search queries performed.

Bias Type	Publications Database	Search Query
Reporting	ACM	[[Abstract: "reporting bias"] OR [Abstract: "reporting biases"]] AND [Abstract: "machine learning"] AND [[Abstract: mitigation] OR [Abstract: mitigating] OR [Abstract: identifying] OR [Abstract: identification] OR [Abstract: addressing]]
	IEEE Xplore	((“Abstract”:reporting bias) OR (“Abstract”:reporting biases)) AND (“Abstract”:machine learning) AND ((“Abstract”:mitigation) OR (“Abstract”:mitigating) OR (“Abstract”:identifying) OR (“Abstract”:identification) OR (“Abstract”:addressing))
	Scopus	TITLE-ABS-KEY (“reporting bias” OR “reporting biases”) AND “machine learning” AND (mitigation OR mitigating OR identifying OR identification OR addressing))
	Science Direct	Title, abstract, keywords: ((“reporting bias” OR “reporting biases”) AND “machine learning” AND (mitigation OR mitigating OR identifying OR addressing))
Selection	ACM	[[Abstract: "selection bias"] OR [Abstract: "selection biases"]] AND [Abstract: "machine learning"] AND [[Abstract: mitigation] OR [Abstract: mitigating] OR [Abstract: identifying] OR [Abstract: identification] OR [Abstract: addressing]]
	IEEE Xplore	((“Abstract”:selection bias) OR (“Abstract”:selection biases)) AND (“Abstract”:machine learning) AND ((“Abstract”:mitigation) OR (“Abstract”:mitigating) OR (“Abstract”:identifying) OR (“Abstract”:identification) OR (“Abstract”:addressing))
	Scopus	TITLE-ABS-KEY (“selection bias” OR “selection biases”) AND “machine learning” AND (mitigation OR mitigating OR identifying OR identification OR addressing))
	Science Direct	Title, abstract, keywords: ((“cognitive bias” OR “cognitive biases”) AND “machine learning” AND (mitigation OR mitigating OR identifying OR addressing))
Cognitive	ACM	[[Abstract: "cognitive bias"] OR [Abstract: "cognitive biases"]] AND [Abstract: "machine learning"] AND [[Abstract: mitigation] OR [Abstract: mitigating] OR [Abstract: identifying] OR [Abstract: identification] OR [Abstract: addressing]]
	IEEE Xplore	((“Abstract”:cognitive bias) OR (“Abstract”:cognitive biases)) AND (“Abstract”:machine learning) AND ((“Abstract”:mitigation) OR (“Abstract”:mitigating) OR (“Abstract”:identifying) OR (“Abstract”:identification) OR (“Abstract”:addressing))
	Scopus	TITLE-ABS-KEY (“cognitive bias” OR “cognitive biases”) AND “machine learning” AND (mitigation OR mitigating OR identifying OR addressing))
	Science Direct	Title, abstract, keywords: ((“cognitive bias” OR “cognitive biases”) AND “machine learning” AND (mitigation OR mitigating OR identifying OR addressing))
Estimators	ACM	[[Abstract: "comparative study"] OR [Abstract: "review"]] AND [Abstract: "machine learning"] AND [Abstract: "algorithm"] AND [Abstract: "bias"]
	IEEE Xplore	((“Abstract”:“comparative study” OR “Abstract”:“review”) AND (“Abstract”:“machine learning”) AND (“Abstract”:“algorithm”) AND (“Abstract”:“bias”))
	Scopus	TITLE-ABS-KEY (“comparative study” AND “machine learning” AND “algorithm” AND “bias”) AND (LIMIT-TO (SUBJAREA, “COMP”))
	Science Direct	Title, abstract, keywords: (“comparative study” AND “machine learning” AND “algorithm” AND “bias”)

Table 2. Cont.

Bias Type	Publications Database	Search Query
Optimizers	ACM	[Abstract: "optimizer"] AND [Abstract: "hyperparameter"] AND [Abstract: "machine learning"] AND [Abstract: "neural network" AND [Abstract: "bias"]]
	IEEE Xplore	((("Abstract":optimizer) AND ("Abstract":hyperparameter) AND ("Abstract":machine learning) AND ("Abstract":neural network) AND ("Abstract":bias)))
	Scopus	TITLE-ABS-KEY ("optimizer" AND "hyperparameter" AND "machine learning" AND "neural network" AND "bias")
	Science Direct	Title, abstract, keywords: ("optimizer" AND "hyperparameter" AND "machine learning" AND "neural network" AND "bias")
Regularization	ACM	[Abstract: "regularization"] AND [Abstract: "machine learning"] AND [Abstract: "bias"] AND [[Abstract: "lasso"] OR [Abstract: "ridge"] OR [Abstract: "elastic net"] OR [Abstract: "data augmentation"] OR [Abstract: "early stopping"] OR [Abstract: "dropout"] OR [Abstract: "weight decay"]]
	IEEE Xplore	((("Abstract":regularization) AND ("Abstract":machine learning) AND ("Abstract":lasso) OR ("Abstract":ridge) OR ("Abstract":elastic net) OR ("Abstract":data augmentation) OR ("Abstract":early stopping) OR ("Abstract":dropout) OR ("Abstract":weight decay)))
	Scopus	TITLE-ABS-KEY ("regularization" AND "machine learning" AND "bias" AND ("lasso" OR "ridge" OR "elastic net" OR "data augmentation" OR "early stopping" OR "dropout" OR "weight decay"))
	Science Direct	Title, abstract, keywords: ("regularization" AND "machine learning" AND "bias" AND ("lasso" OR "ridge" OR "elastic net" OR "data augmentation" OR "early stopping" OR "dropout" OR "weight decay"))

Having retrieved the research studies, a cleaning step took place where the duplicate studies were removed from the list of collected documents. Afterward, this list was further reduced, based on three (3) inclusion criteria (i.e., IC) and one (1) exclusion criterion (i.e., EC). The context of a study should obey the aforementioned IC in order to be considered for this literature review. More specifically, the IC and EC that are taken into consideration are the following:

1. IC#1: The study should contain a specific methodology, either theoretical framework or technical method/algorithm for identifying and/or mitigating bias.
2. IC#2: The study takes algorithmic bias into consideration when comparing different algorithms or optimizers.
3. IC#3: The study explains why a specific algorithm/optimizer/regularization method was chosen.
4. EC#1: The study was not peer reviewed.

IC#1 and EC#1 refer to the studies related to all three (3) categories of bias (i.e., data, algorithms and engineers). IC#2 refers to the studies related to the two (2) subtypes of algorithm bias, namely estimator bias and optimizer bias while IC#3 corresponds to the studies that are related to algorithm bias (i.e., including regularization bias). A visualization of the process followed to select the studies that are analyzed in this manuscript, which was also presented above, is depicted in Figure 2.



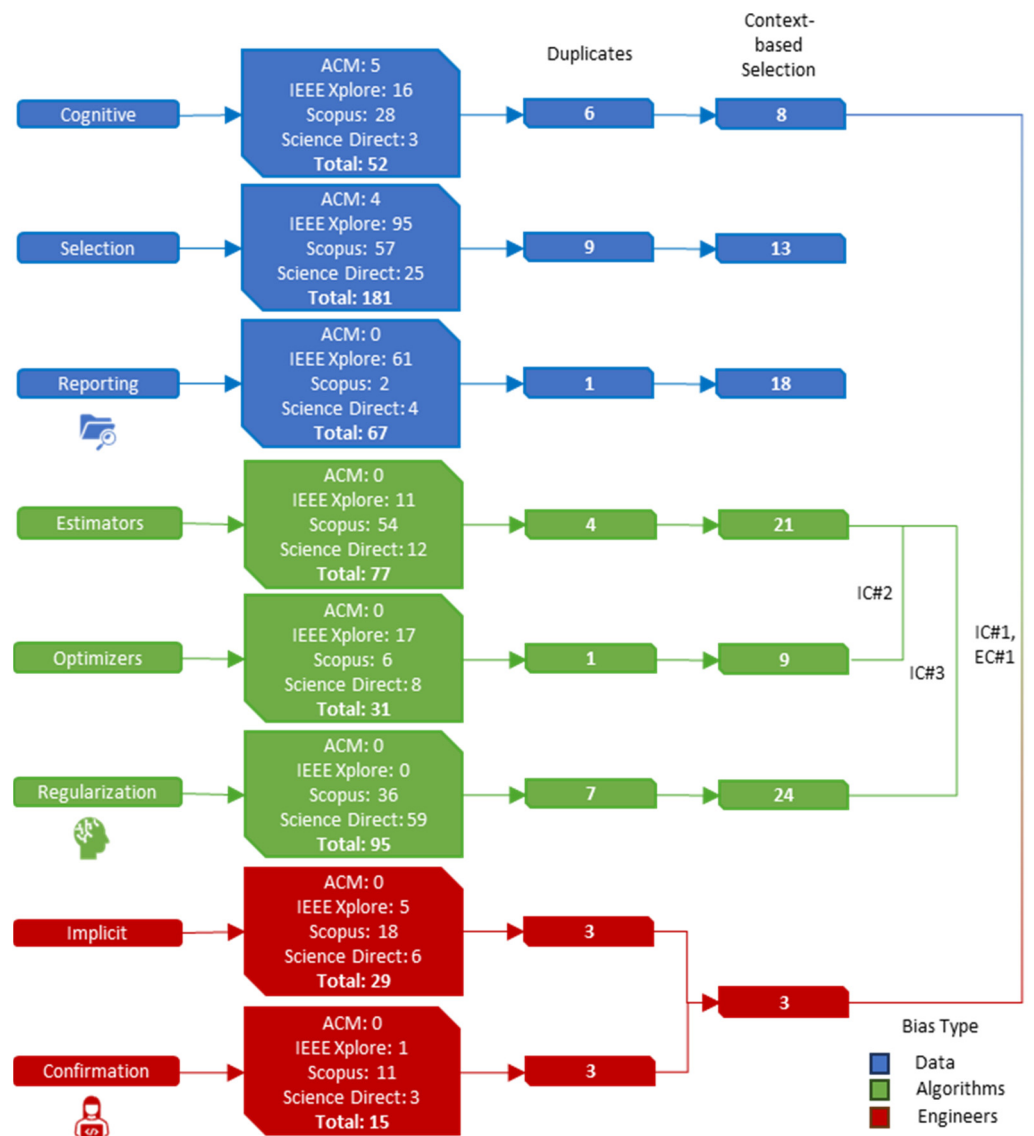


Figure 2. Selection process of relative studies.

### 3. Data Bias

As discussed earlier, one major cause of biased results of ML models is the fact that the data used for training those models are also biased. Data bias is mostly related to the way that data are selected and collected, as well as the nature of the data. As a result, three categories of data bias can be identified, namely reporting bias, selection bias and cognitive bias.

Apart from approaches that exist for dealing with specific categories of data bias which will be analyzed in the following subsections, there are some more scenario-agnostic techniques that can be applied in order to reduce data bias. These techniques mostly refer to imbalanced datasets (i.e., one label occurs more often in the data than the other(s)) and aim to reduce the bias in the datasets by under-sampling or over-sampling [16].

In under-sampling, the number of data instances that belong to the majority class is reduced, whilst in over-sampling, the number of data instances that belong to the minority class is increased [17]. A common approach for performing under-sampling includes the application of a clustering algorithm to cluster the imbalanced dataset and then remove instances of the majority class, based on the cluster to which every instance belongs [18]. Other techniques that can be applied are random under-sampling and Tomek links under-sampling. In random under-sampling, as its name suggests, a random number

of observations to keep is selected [19]. In Tomek links under-sampling, data instances that “overlap” each other are identified, thus removing instances of the majority class that belong to the class “boundaries” [20].

With regard to over-sampling, there exist two main approaches, namely the synthetic minority over-sampling technique (i.e., SMOTE) and adaptive synthetic sampling (i.e., ADASYN). SMOTE first identifies the minority class instances then selects the nearest neighbor and generates a synthetic data sample by joining the minority class instance and the corresponding nearest neighbor [21]. As for the ADASYN, it is similar to SMOTE; however, it generates a different number of samples based on the distribution of the minority class, thus also improving learning performance [22]. What is more, there exist approaches that perform over-sampling on multi-class datasets such as Mahalanobis distance-based over-sampling (i.e., MDO) and adaptive Mahalanobis distance-based over-sampling (i.e., AMDO) that are appropriate for numeric and mixed data, respectively [23].

### 3.1. Cognitive Bias

Cognitive bias is mostly related to the nature of the data. For example, text data from social media posts are quite likely to represent the cognitive bias of the people who wrote those posts. This kind of bias is several times identified in LLMs [24], since they are usually trained, among others, on this kind of data. At this point, it is important to note that this kind of human bias is not related to the bias caused by the ML model builders and evaluators. They are both caused by humans, but the first one is interconnected with the data, whilst the second one is associated with the implementation and evaluation of the ML models.

There have been several attempts to identify cognitive bias in text data such as the one presented in [25]. In this paper, the authors try to automatically detect 188 cognitive biases that are present in the 2016 Cognitive Bias Codex [26] and test its performance to both human and AI-generated text data, showcasing promising results. However, the authors also state that they need more human participants in their study in order to use their responses as the ground truth and establish a better understanding of the approach’s accuracy. In a similar approach that also focuses on text data [27], where the authors try to mitigate cognitive bias that can be found in text data, an ontology-based approach named Automatic Bias Identification (i.e., ABI) is showcased. The main idea of ABI is to consider the decisions made by the users in the past in order to identify whether new, different decisions made by the same users contain cognitive bias. In essence, based on the historical decisions available to the tool, it performs a Boolean classification regarding whether a new decision is biased or not.

In [28], the authors aim to identify four types of cognitive bias in ML models by allowing users to provide feedback to the said models and then examining whether this feedback increases or decreases the variance of future choices made by the ML models. In [29], BiasEye is introduced, which aims to identify and mitigate cognitive bias in text data and, more precisely, text data coming from curriculum vitae (i.e., CVs). BiasEye depends on an ML approach that models individual screening preferences so that it improves information transparency and accessibility, thus enhancing awareness regarding cognitive biases. Similarly, regarding text data from CVs, the authors in [30] propose a three-step approach for mitigating cognitive bias, consisting of pre-process, in-process and post-process steps. More specifically, for the pre-process step, they examine three techniques, namely optimized pre-processing (i.e., OPP), massaging–reweighting–sampling (i.e., MRS) and disparate impact reduction (i.e., DIR). Regarding the in-process step, they utilize adversarial debiasing (i.e., AD) and prejudice remover (i.e., PR). As for the post-process step, they test three techniques namely equalized odds post-processing (i.e., EOP), calibrated equalized odds post-processing (i.e., CEOP) and reject option classification (i.e., ROC). Based on their experiments, they conclude that the combination of DIR, AD and EOP achieves the best results but has higher complexity and computational cost, whereas without the utilization of AD, the accuracy is slightly lower but, still, quite promising. In [31], the integration



of causal analysis results is suggested in order to mitigate cognitive biases. The authors argue that this integration, as well as the utilization of knowledge discovery mechanisms, alongside data-driven methodologies, will assist in avoiding cognitive biases.

In another study [32], the authors also propose an approach to not only identify cognitive bias but also mitigate it, utilizing a module that alters data attributes that cause cognitive bias and then evaluates the results by measuring the KL divergence value. Their approach has been tested in computer vision—related datasets and more specifically in image data, showcasing promising results. In [33], the authors propose a different approach for mitigating cognitive bias in medical LLMs. In their study, they propose “informing” the model that the provided input question may lead to an answer (i.e., model output) that contains some kind of cognitive bias. In this approach, the original training data are not altered in any way, but the model modifies the provided answers based on the assumption that the training data is biased.

A complete list of approaches that have been proposed to identify and/or mitigate cognitive bias can be found in Table 3.

**Table 3.** Approaches to identify and/or mitigate cognitive bias.

ID	Year	Ref.	Domain	Data Category	Type
CB1	2023	[25]	Natural Language Processing (i.e., NLP) (Generic)	Text	Identification
CB2	2023	[27]	NLP-Generic	Text	Identification
CB3	2020	[28]	Generic	Not specific	Identification
CB4	2024	[29]	NLP (Business)	Text	Identification and Mitigation
CB5	2020	[30]	NLP (Business)	Text	Identification and Mitigation
CB6	2024	[31]	Generic	Not specific	Identification and Mitigation
CB7	2022	[32]	Computer Vision	Images	Identification and Mitigation
CB8	2024	[33]	NLP (Health)	Text	Identification and Mitigation

### 3.2. Selection Bias

Another type of bias that can be identified in the data is the selection bias. Selection bias occurs when the data samples that are selected are not reflective of the real-world distribution of the data [34]. Under the umbrella of this kind of bias, there exists a plethora of subtypes of bias such as sampling, attrition, self-selection, survivorship, nonresponse and under-coverage bias [35].

In general, selection bias can be avoided by applying the proper techniques when sampling the population of interest and designing the corresponding study, thus understanding the connection between the data samples and the target population [36]. Except for the initial stages of a study, selection bias can also be addressed later on, when the data has already been collected.

To achieve this, there have been proposed several methods in the literature, mainly statistical ones. Those statistical methods include the delta-method, linear scaling, power transformation of precipitation [37], empirical quantile mapping, gamma quantile mapping and gamma-pareto quantile mapping [38]. Selection bias is quite common in historical climate-related data; thus, the aforementioned methods are greatly used in relevant use cases, such as rainfall-runoff processes [39]. Another approach that aims to identify selection bias in environmental observational data is presented in [40], where the authors introduce two methods. The first one relies on the uniqueness of members of exponential families

over any set of non-zero probabilities. The second method depends on the invariances of the true generating distribution before selection. The authors claim that both approaches generate adequate results for identifying the variables affected by selection bias.

Similarly, in [41] an approach for mitigating selection bias in environment time series data is also presented. It consists of two cross-validation iterations. The outer one is used for identifying the set of variables based on which the corresponding predictor produces the minimum residual sum of squares (i.e., RSS), whilst in the inner iteration a grid search is performed to identify the optimal hyperparameters corresponding to the minimum RSS. Regarding computer vision, three approaches can be found in the literature. The first one is presented in [42] and makes use of a causally regularized logistic regression (i.e., CRLR) model for classifying data that contain agnostic selection bias. The model consists of a weighted logistic loss term and a subtly designed causal regularizer. The second one is showcased in [43], where the authors utilize an adapted triplet loss (i.e., ATL) method. In ATL, a matching loss term reduces the distribution shift between all possible triplets and the selected triplets, thus minimizing selection bias. The third one is presented in [44] and mainly focuses on medical images and utilizes a technique called Recalibrated Feature Compensation (i.e., RFC). RFC uses a recalibrated distribution to augment features for minority groups, by restricting the feature distribution gap between different sample views. With regard to the health domain, an additional five can be found in the literature.

In [45], the selection bias that can be found in electronic health records (i.e., EHRs), which is caused due to missing values can be reduced by implementing an inverse probability weighting (i.e., IPW) adjustment technique. Moreover, the authors utilized chained equations to fill any additional missing features, showcasing that data-cleaning techniques can also assist in mitigating data bias. Data cleaning is also used in [46] to address selection bias. A threshold of 30% is set which means that if a feature has more than 30% missing values, it should be eliminated. The goal of this procedure is to ensure that every feature consists of the approximately same number of missing values. Moreover, since the data come from different hospitals of different sizes, the dataset is clipped in order to contain an equal number of records from every hospital, thus avoiding the domination of large-sized hospitals.

Similarly, in [47] a clique-based causal feature separation (i.e., CCFS) algorithm is presented that theoretically guarantees that either the largest clique or the rest of the causal skeleton is the exact set of all causal features of the outcome. In [48], the authors address the issue of medical question-answering (i.e., Q&A) and propose label resetting in order to avoid the effects of selection bias in medical text data, simultaneously preserving the textual content and the corresponding semantic information. In another health-related approach [49], four propensity methods are presented and tested, including one-to-one matching with replacement, IPW and two methods based on doubly robust (i.e., DR) estimation that combines outcome regression and IPW.

In [50], IPW is also being used and a two-stage process if followed, consisting of a classification and a regression model that uses the reweighted results of the classification. In essence, this procedure enhances underrepresented records in the dataset and reduces the weights of overrepresented records. A group fairness metric for model selection called area delimited between classification (i.e., ABC) is introduced in [51] that is based on item response theory (i.e., IRT) and differential item functioning (i.e., DIF) characteristic curve and aims to identify selection bias, also showcasing promising results.

Apart from the above-mentioned methods, the authors in [52] introduce Imitate (Identify and Mitigate Selection Bias) for mitigating selection bias, which they have also released as a Python module [53]. The idea behind this approach is that the algorithm first calculates the dataset's probability density and then adds generated points to smooth out the density. If the points are concentrated in certain areas and are not widespread, this could be an indication that selection bias is present in the dataset.

However, the Imitate algorithm assumes that there exists only one normally distributed cluster per class in the dataset. To this end, the authors in [54] propose MIMIC (Multi-

IMitate Bias Correction), which utilizes Imitate as a building block, being, however, capable of handling multi-cluster datasets by speculating that the ground-truth data consists of a set of multivariate Gaussians. Even though this is a limiting assumption, MIMIC is able to deal with more datasets than Imitate. In another approach [55], the authors discuss selection bias in recommender systems, which is a common issue in such systems, since they heavily rely on users' historical data [56]. Their proposed approach consists of a data-filling strategy utilizing, irrelevant to the users, items based on temporal visibility.

A complete list of approaches that have been proposed to identify and/or mitigate selection bias can be found in Table 4.

**Table 4.** Approaches to identify and/or mitigate selection bias.

ID	Year	Ref.	Domain	Data Category	Type
SB1	2019	[39]	Environment	Time series	Identification
SB2	2023	[40]	Environment	Observations (numeric)	Identification
SB3	2021	[41]	Environment	Time series	Identification and Mitigation
SB4	2018	[42]	Computer Vision	Images	Identification and Mitigation
SB5	2022	[43]	Computer Vision	Images	Identification and Mitigation
SB6	2023	[44]	Computer Vision (Health)	Images	Identification and Mitigation
SB7	2019	[45]	Health	EHR data (numeric and text)	Identification and Mitigation
SB8	2023	[46]	Health	Mixed (text and numeric)	Identification and Mitigation
SB9	2023	[47]	Health	Numeric	Identification and Mitigation
SB10	2023	[48]	Health	Text (NLP—Q&A)	Identification and Mitigation
SB11	2021	[49]	Health	Mixed (text and numeric)	Identification and Mitigation
SB12	2017	[50]	Not specific	Not specific	Identification and Mitigation
SB13	2023	[51]	Mixed (text and numeric)	Several (binary classification)	Identification

### 3.3. Reporting Bias

An additional type of bias that can be encountered in the data is the reporting bias. Reporting bias is quite common in Language Models (LMs) [57] and a plethora of studies have concluded that LMs such as RoBERTa and GPT-2 amplify reporting bias, due to the data that they have been trained on [58].

To address this issue, the authors in [59] propose a neural logic-based Soft Logic Enhanced Event Temporal Reasoning (SLEER) model in order to mitigate the reporting bias originating from the Temporal Common Sense (TCS) knowledge that is extracted from free-form text. In another paper, the authors focus on text-image datasets and the reporting bias that might occur in those cases [60]. They propose the bimodal augmentation (BiAug) approach that generates pairs of object-attributes to diminish the over-representation of recurrent patterns. However, the generation of data (i.e., pairs of object-attributes) is quite expensive in terms of computational resources due to the utilization of LMs. The authors suggest that this challenge can be mitigated by utilizing techniques such as parallel processing.

In a similar approach focusing on image data [61], the authors suggest that the reporting bias can be removed by using label frequencies. Those frequencies, which are the per-class fraction of labeled and positive examples found in all positive examples, are being

estimated using Dynamic Label Frequency Estimation (DLFE); thus, an increased accuracy of the debiasing process is observed.

Similarly, in [62], it is suggested that a new variable should be added in the training dataset describing what is shown in every image, except for the image's label. In that way, this technique leverages both the aspects of human performance and the algorithmic understanding, thus showcasing quite promising results.

In another approach [63], where the authors focus on the task of facial recognition, the authors experiment with five algorithms namely, logistic regression (i.e., LR), linear discriminant analysis (i.e., LDA), k-nearest neighbors (i.e., KNN), support vector machines (i.e., SVM) and decision trees (i.e., DT). Based on their results, DT seems to better address reporting bias that is present in the corresponding data. Regarding cybersecurity, two approaches can be found in the literature that focus on threat classification and malware detection, respectively. The first approach makes use of four techniques for mitigating reporting bias in the data, namely covariate shift, kernel mean matching, Kullback–Liebler importance estimation procedure (i.e., KLIEP) and relative density ratio estimation [64]. In the second approach, the authors split the private and public datasets that they have at their disposal in different ways prior to training and testing the implemented convolutional neural network (i.e., CNN). The training and testing of the model occur in a disjoint way, using different timescales, thus mitigating the reporting bias of the training data [65]. In [66], the authors utilize a statistical technique called repeated measures correlation (i.e., RMCorr) to avoid reporting bias from health-related data, by using multiple samples from each subject.

Similarly, the authors in [67] mainly focus on the data collection phase in order to avoid reporting bias by explicitly adjusting the survey responses related to the intention of patients to vaccinate. Those responses are then used to train a random forest (i.e., RF) algorithm that predicts whether a specific patient will be vaccinated on time. The approach presented in [68], also focuses on the data collection phase to mitigate potential reporting bias. In this case, the authors utilize specific software (<https://acasillc.com/acasi.htm>, 22 March 2024) called audio computer-assisted self-interviewing (i.e., ACASI) and computer-assisted personal interviewing (i.e., CAPI) to provide a secure, private and confidential environment, thus increasing the level of answers' honesty to questions related to sensitive behaviors such as illicit drug use. Focusing on an entirely different domain and trying to estimate the dropout rate in college institutions, the authors in [69] propose Disc-Less (Discrimination-Less) which is based on the advanced process optimizer (i.e., ADOPT).

In a similar case [70], where the authors try to predict whether a university student will fail a class, they compare three algorithms which are LR, Rawlsian and cooperative contextual bandits (i.e., CBB). LR has the highest accuracy but is prone to bias. Rawlsian is a fairness algorithm but in this specific case fails to remove bias. Finally, CBB consists of two Gradient Contextual Bandits, which not only mitigate reporting bias that is present in the data but are also easier to implement in contrast to other similar baseline models such as Learning Fair Representation (i.e., LFR) and Adversarial Learned Fair Representation (i.e., ALFR). In [71], a complete toolkit is presented that allows the users to compare thirteen ML algorithms and utilize twelve (12) evaluation metrics, as well as perform a significance test to reduce reporting bias. The results of this toolkit are very promising; however, it has been utilized only for aerospace prognostic data. In [72], Bosco is introduced to deal with imbalanced biology data. The main rule that Bosco follows is assigning samples in the major group with different important scores, thus mitigating potential bias in the data.

As for time series data, the authors in [73] compared two known techniques, namely quantile mapping and power transformation to correct the data coming from several different regional climate models. Their results showcased that power transformation outperformed quantile mapping, since the corrected data that power transformation generated, enhanced the accuracy of the model that was trained based on them. In [74], the authors suggest that Privacy-Preserving Decentralized Intelligence can also be utilized to avoid bias by training ML models over a network of autonomous entities without the training

data being exposed. Except for the model training, data analysis can also take place in a distributed manner, which could also help mitigate bias.

In another approach [75], the authors focus on financial data and the mitigation of the corresponding reporting bias that might occur, either due to human error or deliberately as part of a fraud. In this case, a hybrid method is showcased that combines the input of human experts on the corresponding field, as well as the utilization of Shannon entropy analysis to evaluate the information that is provided by the experts. Finally, in order to identify reporting bias in social media data, the authors in [76], compare a set of traditional ML algorithms, namely, Naïve Bayes (i.e., NB), LR and SVM. SVM managed to achieve the highest accuracy in identifying the bias that is present in the data, showcasing that it can be a useful tool for identifying reporting bias in text data.

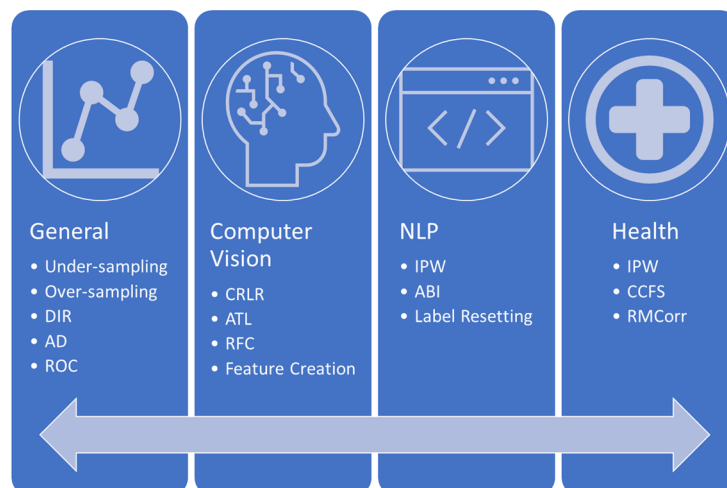
A complete list of approaches that have been proposed to identify and/or mitigate selection bias can be found in Table 5.

**Table 5.** Approaches to identify and/or mitigate reporting bias.

ID	Year	Ref.	Domain	Data Category	Type
RB1	2022	[59]	Not specific	Text	Identification and Mitigation
RB2	2023	[60]	Computer Vision	Text and Images	Identification and Mitigation
RB3	2021	[61]	Computer Vision	Images	Identification and Mitigation
RB4	2016	[62]	Computer Vision	Images	Identification and Mitigation
RB5	2021	[63]	Facial Recognition	Images	Identification and Mitigation
RB6	2018	[64]	Security	Numeric	Identification and Mitigation
RB7	2019	[65]	Security (Malware Detection)	Numeric	Identification and Mitigation
RB8	2023	[66]	Health	Numeric	Identification and Mitigation
RB9	2023	[67]	Health	Numeric	Identification and Mitigation
RB10	2020	[68]	Health	Numeric	Identification and Mitigation
RB11	2023	[69]	Education	Numeric	Identification and Mitigation
RB12	2020	[70]	Education	Numeric	Identification and Mitigation
RB13	2019	[71]	Aerospace	Prognostic Data	Identification and Mitigation
RB14	2017	[72]	Biology	Mixed	Identification and Mitigation
RB15	2023	[73]	Environment	Time series data	Identification and Mitigation
RB16	2023	[74]	Generic	Mixed	Identification and Mitigation
RB17	2021	[75]	Finance	Numeric	Identification and Mitigation
RB18	2018	[76]	Social media	Text	Identification

### 3.4. Common Mitigation Techniques for Data Bias

According to the above-mentioned findings, there exist some common issues with regard to data bias, as well as corresponding techniques that can be applied to mitigate it. More specifically, as stated in the beginning of this section, common techniques that can be applied to address data bias, regardless of the use case, are under-sampling and over-sampling. As for use-case-specific approaches, major issues regarding data bias are mostly common in NLP, computer vision and the health domain. The relatively common domains that are affected from data bias and the corresponding solutions that have been proposed in the literature and presented above are depicted in Figure 3.



**Figure 3.** Relatively common domains and solutions with regard to data bias.

By comparing the above-mentioned techniques, in terms of the domain that they are applied to, it is noticeable that methods like under-sampling, over-sampling, DIR, AD and ROC are interesting candidates that can be used for mitigating data bias, regardless of the domain. Moreover, IPW is also a useful technique when it comes to text data and can be applied to several domains as well, given the fact that the corresponding data are text data. With regard to image data, there exist specific techniques that are appropriate for this kind of data which can also be applied to several domains, like health, considering that the corresponding data are image data.

#### 4. Algorithm Bias

##### 4.1. Estimators

Except for the bias that originates from the data and described above, bias can also occur due to the ML models that are being utilized. To begin with, the most obvious reason that leads to bias is the selection of an algorithm (i.e., estimator) that is not suitable for the given task (e.g., using a linear model to provide prediction on non-linear data). Another reason is that, given a certain task (e.g., classification), an algorithm might outperform others in a specific use case but fall behind in other use cases, due to the nature of the corresponding data. To this end, various comparative studies have been published that compare several ML algorithms and try to propose the most appropriate ones for specific use cases by minimizing the bias of the results and thus maximizing the accuracy of the predictions.

Prior to presenting the said comparative studies, a brief description of every estimator that will be mentioned later on should be provided so that the rest of the context will be more understandable by the readers of this manuscript. The names of the algorithms that are compared in the comparative studies that are analyzed below, as well as a brief description of the said algorithms, are shown in Table 6. The algorithms are sorted alphabetically.

**Table 6.** High-level description of the algorithms that are tested in the corresponding literature.

Algorithm (Estimator) Name	Abbreviation	Description
Autoregressive Moving Average	ARIMA	Mostly used in time series forecasting, where the current value of the series can be explained as a function of past values [77]
Adaptive Neuro-Fuzzy Inference System	ANFIS	Type of a Neural Network that is based on the Takagi-Sugeno fuzzy system [78]
Bee Colony Optimization	BCO	System that consists of multiple agents for solving complex optimization problems [79]



Table 6. Cont.

Algorithm (Estimator) Name	Abbreviation	Description
Bidirectional Long Short-Term Memory Neural Network	Bi-LSTM NN	Specific type of LSTM neural network that enables additional training by traversing the input data [80]
Cascade Support Vector Machine	C-SVM	Ensemble ML technique that consists of several SVMs stacked in a cascade [81]
C4.5 Classifier	C4.5	Type of a decision tree classifier [82]
Classification and regression tree	CART	Type of a decision tree algorithm that can be used for both classification and regression tasks [83]
CNN-Bi-LSTM	CNN-Bi-LSTM	Type of a Bi-LSTM NN that also utilizes a convolutional layer [84]
Conditional Random Fields	CRF	Statistical modeling method that is mostly used in pattern recognition and ML for structured predictions, since it models predictions into graphs [85]
Convolution Long Short-Term Memory	CLSTM	Type of an LSTM NN that also utilizes a convolutional layer [86]
Convolutional Neural Network	CNN	A type of feed-forward NN that consists of at least one convolutional layer and is widely used in computer vision tasks [87]
Cox proportional hazards model	Cox regression	Type of a survival model. That kind of models try to associate time prior to one event happening with one or more covariates [88]
Cubist	Cubist	Rule-based model that contains a tree whose final leaves use linear regression models [89]
Decision Trees	DT	Fundamental ML algorithm that uses a tree-like model of decisions and can be used for both regression and classification tasks [90]
Deep Neural Network	DNN	A type of an NN that has multiple hidden layers between the input and the output [91]
Double Deep Q-Network	DDQN	Consists of two separate Q-networks that are a type of reinforcement learning algorithms [92]
Exponential smoothing methods	ESM	Method used in time series forecasting that utilizes an exponentially weighted average of past values to predict a future value [93]
Extra Trees	ET	An ensemble ML approach that utilizes multiple randomized decision trees [94]
Extreme Gradient Boosting	XGBoost	Iteratively combines predictions of multiple individual models, usually DTs [95]
Gene Expression Programming	GEP	Type of evolutionary algorithm that consist of complex trees that adapt and alter their structures [96]
Gradient Boosted Regression Trees	GBRT	Type of additive model that makes predictions by combining decisions from a set of other models [97]
Gated Recurrent Unit Networks	GRU	Type of Recurrent Neural Network, similar to LSTM, that utilize less parameters, thus having less computational cost [98]
k-Nearest Neighbors	KNN	A fundamental ML algorithm used for classification and regression based on proximity of data points [99]
Light Gradient Boosting Machine	LightGBM	Ensemble method that is based on gradient boosting [100]
Linear Discriminant Analysis	LDA	Supervised ML algorithm for classification that aims to identify linear set of features that identify classes into a dataset [101]
Linear Regression	LinearR	Fundamental algorithm that estimates the linear relationship between two variables [102]
Linear Support Vector Classification	LinearSVC	Subtype of SVM used for perfectly linear data [103]
Logistic Regression	LogR	Used for estimating the parameters of logistic models [104]
Long-Short Term Memory	LSTM	Type of recurrent neural network that is particularly useful for time series forecasting, since it is capable to “remember” [105]

Table 6. Cont.

Algorithm (Estimator) Name	Abbreviation	Description
Multi-Layer Perceptron neural network	MLP	Subtype of a DNN [106]
Multinomial Naive Bayes	MNB	Probabilistic classifier that is based on the Bayes' theorem and focuses on calculation of text data's distribution [107]
Neural Network	NN	Structure that is made of artificial neurons that receive signals from other connected neurons and create an output that can be forwarded to other neurons [108]
Random Forest	RF	Fundamental ML algorithm that makes use of the output of multiple decision trees for classification, regression and other tasks [109]
Squeaky Wheel Optimization	SWO	An optimization technique based on a greedy algorithm [110]
Seasonal and Trend decomposition using Loess	STL	Method used in time series where linear regression models are applied to decompose a time series into separate components [111]
Support Vector Machines	SVM	Classification algorithm that finds an optimal line or hyperplane to maximize the distance between each class [112]
Temporal Difference (lambda)	TD	Reinforcement learning method that share common characteristics with Monte Carlo method and Dynamic Programming methods [113]
Term Frequency Inverse Document Frequency of records	TF-IDF	A measure calculating the relevancy of a word in a series of words or entire corpus [114]
Value-Biased Stochastic Search	VBSS	Randomization method that can be used for determining stochastic bias [115]
Vector Autoregression	VAR	Model that can be used for multiple time series that influence each other [116]

Having presented the algorithms that will be discussed in this section, the corresponding comparative studies are now going to be analyzed. Starting from the agricultural domain, the authors in [117] compare three algorithms to estimate soil organic carbon (i.e., SOC), namely RF, SVM and Cubist. Based on the results of the experiments, an ensemble method that utilized all three algorithms was proposed, since it reduced the overall bias when modeling SOC, regardless of the different spectral measurement conditions. In [118], the authors compare several algorithms from reinforcement learning, including SWO, VBSS, BCO and TD, for the scheduling problem. They conclude that SWO performed better since it was particularly designed to solve that kind of problem.

Regarding biological data used to predict bacterial heterotrophic fluxomics, the authors in [119] test SVM, KNN and DT and conclude that SVM is the most appropriate one in this use case since it achieves the highest accuracy.

As for the business domain, the authors in [120] test the MNB and a CLSTM NN for the classification of clients' reviews. Their results showcase that CLSTM NN is more appropriate for this classification task because the MNB produces biased results since it is learning a single class and neglects the other ones because the training data are imbalanced. In a similar approach [121], the authors compare RF, SVM and a DNN for predicting whether a machine in an assembly line will fail, based on its kinematic, dynamic, electrical current, and temperature data. Even though SVM reduced model overfitting and bias, it did not achieve the desired accuracy whilst the DNN not only improved the accuracy but also produced insignificant bias.

In [122], where the authors handle time series data of marine systems the authors experiment on a set of algorithms and methods including STL decomposition, ESM, ARIMA, LinearR, KNN, SVM, VAR, DT and ensemble methods that combine the previously mentioned methods. The authors suggest that the ensemble methods should be avoided since they have a higher computational cost. They also argue that for this specific use case, VAR should be used when many missing values are present in the time series, whilst ARIMA should be chosen if there are no missing values. In a computer vision problem [123], where

data from remote sensing are available (e.g., data coming from satellites), the authors test RF, ET and GBRT. Out of the three, ET performed the best in terms of accuracy, while GBRT performed the worst. In [124], XGBoost, LightGBM and GEP are studied in the context of estimating the quality of steel-fiber-reinforced concrete. Even though all of them generate adequate predictions, GEP is selected for forecasting the shear strength considering the most important factors, such as the beam depth effect. The authors in [125] experiment with CART, ANFIS, SVM and an MLP NN for performing time series forecasting using hydro-meteorological data. They conclude that the CART algorithm can be a very useful tool since it generates the most promising results in terms of the Nash–Sutcliffe model efficiency coefficient (i.e., NSE), Kling–Gupta efficiency (i.e., KGE) and Percent Bias (i.e., PBias).

In another approach related to the environment [126], a fully connected NN, a CNN and an LSTM NN are trained on seismic data and the results showcase that the CNN and the LSTM NN were extremely effective both in terms of Mean Square Error (i.e., MSE) and statistical bias.

In [127], the authors utilize a set of algorithms to predict whether an account holder will be able to repay financial loans in time. Those algorithms include DDQN, SVM, KNN, RF and XGBoost. Based on the results of this comparative study, KNN outperformed all the other algorithms in terms of prediction accuracy, whilst DDQN provided interesting results, paving the way for the application of other reinforcement learning techniques in similar use cases.

With regard to the health domain, there have been several comparative studies that experiment on a diverse set of algorithms and datasets. Starting from [128], the authors compare LogR, an NN, SGD, RF, NB, KNN and DT by training them on a variety of open-source datasets concluding that LogR provided the best results based on a set of evaluation metrics that include, accuracy, precision, recall and F1-Score. In [129], the authors compare LinearSVC, LogR, MNB and RF in the context of identifying stereotypical patient characteristics in clinical science vignettes, concluding that LogR when combined with TF-IDF, can be quite useful for such use cases. In [130], the authors use Cox regression and an NN to predict the outcome of a COVID-19 infection. Even though both achieved high accuracy, the NN had significantly greater discriminatory ability. SVM, an NN and NB are tested in the context of [131], with regard to detecting leukemia and is showcased that SVM produced superior results in terms of classification accuracy. In [132], the authors make use of twenty-three (23) medical datasets and experiment with an NN, a DNN, C4.5, NB, KNN, LogR and SVM in order to find the most appropriate algorithm for medical datasets. Based on their results, SVM is the best-performing algorithm for medical datasets. Similarly, in [133] LogR, DT, RF and an NN are tested. NN had a greater accuracy but for a single class since it was affected by the bias present in the dataset. Taking this into consideration, the authors argue that RF performed the best with regard to all the classes present in the dataset. In [134], SVM and DT are compared. In this case, DT is slightly more accurate than SVM for diagnosing soft tissue tumors; however, it is more sensitive to the total number of parameters. As for medical images, the authors in [135] compare a non-recurrent NN and an LSTM NN, concluding that the LSTM-based approach improves the prediction accuracy, and the F1-Score compared to its non-recurrent counterpart. As for sentiment analysis, the authors in [136] experiment on a vast set of NN-based architectures, such as CNN-Bi-LSTM, LSTM, Bi-LSTM, CNN and GRU, as well as non-NN-based algorithms such as RF and SVM. According to their results, CNN-Bi-LSTM outperformed all the other algorithms in terms of accuracy. Lastly, with regard to sensor data coming from smart homes for automatic activity recognition, the authors in [137] experiment on C-SVM, CRF and LDA by utilizing an imbalanced dataset. According to this study, CRF is more sensitive to overfitting on a dominant class, whilst SVM outperforms the other two with respect to class accuracy, making it a feasible method for this specific use case.

A complete list of comparative studies of algorithms that have been conducted and take bias into consideration can be found in Table 7. The comparative studies are sorted alphabetically based on the domain that each study refers to.

**Table 7.** Complete list of comparative studies of ML algorithms (estimators).

ID	Year	Ref.	Domain	Data Category	Estimators	Most Suitable Estimator
CS1	2022	[117]	Agriculture	Spectra Data	RF, SVM, Cubist	Ensemble Method
CS2	2009	[118]	Automobiles	Mixed (text–numeric)	SWO, VBSS, BCO, TD	TS and SWO
CS3	2016	[119]	Biology	Numeric	SVM, KNN, DT	SVM
CS4	2020	[120]	Business	Text (reviews)	MNB, CLSTM	CLSTM
CS5	2022	[121]	Business	Numeric	RF, SVM, DNN	DNN
CS6	2020	[122]	Business	Time series	STL decomposition, ESM, ARIMA, LinearR, KNN, SVR, VAR, DT	VAR or ARIMA (based on missing values presence)
CS7	2015	[123]	Computer Vision	Remote Sensing	RF, ET, GBRT	ET
CS8	2024	[124]	Construction	Numeric	XGBoost, LightGBM, GEP	GEP
CS9	2018	[125]	Environment	Time series	CART, ANFIS, MLP NN	CART
CS10	2021	[126]	Environement	Seismic data	Dense NN, CNN, LSTM	CNN and LSTM
CS11	2022	[127]	Financial	Numeric	DDQN, SVM, KNN, RF, XGBoost	KNN
CS12	2022	[128]	Health	Numeric	LogR, NN, SGD, RF, NB, KNN, DT	LogR
CS13	2021	[129]	Health	Text	LinearSVC, LogR, MNB, RF	LogR with TF-IDF
CS14	2020	[130]	Health	Numeric	Cox regression, NN	NN
CS15	2017	[131]	Health	Numeric	SVM, NN, NB	SVM
CS16	2021	[132]	Health	Numeric	NN, C4.5 Classifier, NB, KNN, Logistic Classifier, SVM, DNN.	SVM
SC17	2020	[133]	Health	Numeric	LogR, DT, RF, NN	RF
CS18	2022	[134]	Health	Numeric	SVM, DT	SVM
CS19	2021	[135]	Health	Images	Simple NN, LSTM	LSTM
CS20	2023	[136]	Sentiment Analysis	Text	CNN-Bi-LSTM, LSTM, Bi-LSTM, CNN, GRU, RF, SVM	CNN-Bi-LSTM
CS21	2012	[137]	Smart Homes	Sensor Data	C-SVM, CRF, LDA	C-SVM

In the literature, there also exist algorithms that have been developed specifically for bias mitigation and thus can assist in reducing bias that is caused by an estimator like the ones that are presented above. An indicative example of such an approach is adversarial debiasing. Adversarial debiasing is based on adversarial learning and suggests the simultaneous training of a predictor and a discriminator where the first one aims to predict the target variable accurately, whilst the latter one aims to predict the protected variable (e.g., gender). The goal of adversarial debiasing is maximizing the ability of the predictor to predict the target variable while simultaneously minimizing the ability of the discriminator to predict the protected attribute based on the predictions made by the predictor [138]. What is more, a federated version of the aforementioned approach has been proposed in [139] which not only deals with privacy concerns related to the training data but also achieves almost identical performance with the centralized version of it.

Another mechanism that can help in tackling bias when training an estimator and thus, ensuring fairness, are the fairness constraints. A fairness constraint prevents a classifier from outputting predictions that correlate with a protected/sensitive attribute that is present in the data [140]. There exist numerous related approaches, including the one presented in [141], where the authors utilize an intuitive measure of decision boundary (un)fairness to implement fairness constraints while using datasets with multiple sensitive attributes. Similarly, the approaches that are presented in [142,143] are also able to handle

multiple sensitive attributes but do not address the disparate impact's business necessity clause [144].

Furthermore, in the literature, there exists a type of neural network architecture named variational autoencoders which are able to achieve subgroup demographic parity with regard to multiple sensitive attributes, thus reducing bias [145]. Moreover, other approaches that are used in the literature for bias mitigation are contrastive learning [146] and neural style transfer [147]. In the first one, contrastive information estimators are utilized to control the parity of an estimator by limiting the mutual information between representations and protected attributes of a dataset. The second one is an approach for performing image-to-image translation which, in the context of dealing with bias and improving fairness, can be used for mapping from an input domain to a fair target domain.

#### 4.2. Optimizers

Aside from the selection of the appropriate estimator (i.e., algorithm), the utmost care should be taken when selecting optimizers and types of regularization, since all of them can contribute to biased algorithmic decisions, regardless of the absence or presence of bias in the input data [148]. Regarding optimizers, those are algorithms that are utilized in neural networks to change their attributes, such as weights and learning rate, thus minimizing the losses. There have been proposed several optimizers in the literature, from which the most widely used are Stochastic Gradient Descent (SGD) and its variant Stochastic Gradient Descent with Gradient Clipping (SGD with GC), Momentum and its variant Nesterov, Adan (Adaptive Nesterov) and AdaPlus, AdaGrad (Adaptive Gradient), its variation Adadelta, RMSProp (Root Mean Square Propagation) and its variants NRMSProp and SMORMS3 (Squared Mean Over Root Mean Squared Cubed), as well as Adam (Adaptive Moment Estimation) and its variants Nadam, AdaMax and AdamW (Adam with decoupled Weight Decay).

SGD is used for unconstrained optimization problems. It is preferred in cases where there are requirements of low storage space and fast computation speed, and the data might be non-stationary or noisy [149]. However, SGD is one of the most basic optimizers [150]. Moreover, the selection of the learning rate is not an easy task and, if not tuned carefully, it might lead to not ensuring convergence [151]. In order to avoid the slow convergence issue of standard SGD, SGD with GC has been proposed [152]. The key difference between the two optimizers is that in the case of SGD with GC, the gradients are clipped if their norm exceeds a specified threshold, prior to updating the parameters of the model using (1). By doing so, the stability of SGD is enhanced, and convergence is ensured in more cases [153].

Momentum is a type of optimization technique that "accelerates" the training process of neural networks, being first studied in [154]. In contrast to the above-mentioned gradient descend optimizers, this technique does not directly update the weights of an ML model but rather introduces a new term named the "momentum term" that will update the weights by calculating the moving average of the gradients, thus guiding the search direction of the optimizer [155]. The momentum technique has been utilized in the SGD optimizer described above resulting in the variant of SGD with momentum. SGD with momentum works faster and generally performs better than SGD without momentum [156]. As mentioned above, a variant of the Momentum technique is the Nesterov Momentum [157]. In Nesterov Momentum, the main idea is that the "momentum term" mentioned previously can be utilized for the prediction of the next weights' location, thus allowing the optimizer to take larger steps while trying to ensure convergence [158]. A recently proposed optimizer that utilizes the Nesterov Momentum is presented in [159] and is named Adan. Adan has been tested in a variety of use cases, including Deep Learning (DL) related tasks such as NLP and CV, surpassing other currently available optimizers. The source code of the optimizer is available in [160]. Another optimizer that utilizes the Nesterov Momentum and combines the advantages of other optimizers such as AdamW and Nadam is called AdaPlus [161]. AdaPlus has been evaluated in CV-related tasks by being utilized in CNNs



and being compared with other state-of-the-art optimizers, showcasing promising results. The source code of the AdaPlus can be found in [162].

Another optimizer that is widely used in ML and DL is the AdaGrad [163]. AdaGrad is used for gradient-based optimization and the main idea behind AdaGrad is that it adjusts the learning rate per feature, based on the feature's updates. If the feature is being updated often, this means that it frequently occurs in the dataset and the Adagrad optimizer assigns a smaller learning rate. On the other hand, if the feature is not being updated often, this means that this feature is infrequent in the dataset and the AdaGrad optimizer assigns a high learning rate for the corresponding parameters. This makes AdaGrad an appropriate choice when dealing with sparse data [164]. However, even though AdaGrad utilizes adaptive learning rates, as explained above, it is sensitive to the global learning rate that is being initialized at the beginning of the optimization and which may result in not arriving at the desired local optima [165]. In order to address the aforementioned issue, other variants of the AdaGrad have been proposed in the literature, including Adadelta. The main difference between AdaGrad and Adadelta is that the latter does not require to manually set the learning rate [166]. More specifically, Adadelta utilizes "delta updates" in which it calculates the ratio of the root mean squared (RMS) of past gradients and the RMS of the past updates so that it can adjust the learning rate. As a result, Adadelta can potentially outperform AdaGrad in use cases where the latter fails to reach the desired solution [167–169].

Another variation of AdaGrad is RMSProp. The main idea behind RMSProp, which is also the key difference with AdaGrad, is that a weight's learning rate is divided by a running average of the magnitudes of the recent gradients of the aforementioned weight [170,171]. In [172], the authors propose a variant of RMSProp called NRMSProp that utilizes the Nesterov Momentum. NRMSProp is capable of convergence quicker than RMSProp without adding too much complexity as depicted in the experiments that the authors carried out. Another variation of RMSProp is SMORMS3 [173]. SMORMS3 is particularly useful in Deep Neural Networks (DNNs), where the gradients have a high variance and it might prevent the learning rate from becoming too small, which could potentially slow down the optimization process [174].

Another optimizer that has also been proposed in the literature is Adam. Adam is a combination of AdaGrad and RMSProp and thus is able to work with sparse gradients and does not need a stationary objective [175]. It has very little memory requirements since it only requires first-order gradients. The authors that proposed Adam also proposed in the same manuscript AdaMax, which is a variant of Adam based on the infinity norm. Another variant of Adam is Adam with decoupled Weight Decay (AdamW). In AdamW a decay rate is introduced in order to insert a regularization based on the decay of weights during the process of the optimization [176].

In the literature, there have been specific studies that experiment on a set of optimizers to find the most appropriate ones for the corresponding use cases, whilst also taking the reduction in bias into consideration.

More specifically, the authors in [177] propose an NN architecture to estimate crop water requirements. In order to minimize the bias of the approach, they test the SGD, RMSProp and Adam optimizers. According to their results, they select SGD since it increases the accuracy of the NN whilst it is also quite simple to implement. In a similar approach [178], where the authors aim to predict frosts in southern Brazil, they test SGD and Adam for their NN. They observe that experiments using the ADAM optimizer present greater variability and slightly lower accuracy than those using SGD.

Adam and SGD are also compared in [179], where the authors develop an NN for human resources analytics and, more specifically, employee turnover. Based on the accuracy of the predictions, Adam is the optimal optimizer.

In another approach [180], where electricity consumption needs to be predicted, the authors experiment with Adagrad, Adadelta, RMSProp, SGD, Nadam and Adam optimizers in their NN architecture. According to their results, for the given architecture, which is



an LSTM, and for the given dataset, the SGD optimizer is the most suitable one. In [181], the authors aim to assess the Captcha vulnerability. They develop an NN architecture and experiment with Nadam, Adam and Adamax. They conclude that Nadam is the optimal optimizer since it achieves the highest accuracy. In a similar approach [182] for facial expression recognition, the authors test Adam and SGD for their implemented CNN. They conclude that Adam is the appropriate optimizer since it allows CNN to generalize better when dealing with unseen data.

Both those two optimizers are also tested in [183], alongside Adagrad, and RMSProp in order to be used in an NN that is trained on diabetic data. In this case, RMSProp is the preferred optimizer since it achieves the highest accuracy in the least possible time. In a similar approach [184] that focuses on epilepsy detection, SGD is compared with RMSProp and it is shown that SGD achieves the highest accuracy.

Lastly, in [185] the authors develop an LSTM NN for network-based anomaly detection and experiment with seven optimizers, namely Adam, Adagrad, Adamax, Ftrl, Nadam, RMSProp and SGD. Their results define Nadam as the optimal optimizer since it achieves the highest accuracy.

A complete list of comparative studies of optimizers that have been conducted and that also take bias into consideration can be found in Table 8.

**Table 8.** Complete list of comparative studies of optimizers.

ID	Year	Ref.	Domain	Data Category	Optimizers	Most Suitable Optimizer
OCS1	2021	[177]	Agriculture	Numeric	SGD, RMSprop, Adam	SGD
OCS2	2023	[178]	Agriculture	Time Series Data	Adam, SGD	SGD
OCS3	2024	[179]	Business	Numeric	Adam, SGD	Adam
OCS4	2022	[180]	Civil	Numeric	Adagrad, Adadelta, RMSProp, SGD, Nadam, Adam	SGD
OCS5	2021	[181]	Computer Vision	Images	Nadam, Adam, Adamax	Nadam
OCS6	2021	[182]	Computer Vision	Images	Adam, SGD	Adam
OCS7	2021	[183]	Health (diabetes)	Numeric	SGD, Adagrad, Adam, RMSProp	RMSProp
OCS8	2023	[184]	Health	Numeric	SGD, RMSProp	SGD
OCS9	2023	[185]	Security	Numeric	Adam, Adagrad, Adamax, Ftrl, Nadam, RMSProp, SGD	Nadam

Based on the above table, it would be useful to compare the selected optimizer per approach and understand when each one of them is more appropriate to be used, thus minimizing the bias introduced into the model. In general, the Adam optimizer converges faster and has fewer memory requirements than RMSProp, whilst SGD converges slower. However, SGD converges to optimal solutions and is able to better generalize than Adam [186]. As for Nadam, which, as mentioned previously, is a variant of Adam, it converges slightly faster than Adam, thus resulting in less training time. The above-mentioned optimizers are also summarized in Table 9.

**Table 9.** Recommended usage of Adam, RMSProp, SGD and Nadam.

Optimizer	Recommended Usage
Adam	Training time needs to be reduced
RMSProp	Memory requirements are not important
SGD	There is no time constraint
Nadam	Adam does not produce sufficient results

Selecting the appropriate optimizer is not an easy or straightforward process, and excessive attention should be paid to this matter in order to minimize the bias that can

be introduced. To achieve the above-mentioned goal, it is generally a good practice to experiment with a variety of optimizers based on the use case requirements, such as training time and computational resources, and monitor the training parameters to decide which one should be selected.

#### 4.3. Regularization

As also mentioned earlier, apart from the selection of optimizers, equally great care should be taken when selecting a regularization method in order to avoid biased results from the algorithms. Regularization is a set of techniques and methods that are applied in order to deal with overfitting, which is a frequent phenomenon in ML where a model fits very well in the training data but fails to generalize when fed unseen data [187]. Regularization methods are known to suffer from a certain bias since they need to increase the weight of the regularization term [188]. At this point, the bias–variance tradeoff should be introduced and explained. As mentioned above, in ML, bias measures the difference between the predicted values and the ground true. Variance measures the difference between the predicted values across various applications of the model. Increasing variance means that the model fails to predict accurately on unseen data [189]. In other words, high variance indicates an error during testing and validation, whilst high bias indicates an error during training. Simultaneously decreasing bias and variance is not always possible, which leads to the adaptation of regularization techniques that aim to decrease a model's variance at the cost of increased bias [190]. However, recent surveys suggest that the above-mentioned is not entirely true since it is possible to decrease both bias and variance [191].

Towards this direction, there have been proposed several regularization techniques in the literature. First of all, with regard to ML problems where linear models are utilized, three different types of regularization can be applied, namely Lasso (L1) regression, Ridge regression (L2) and elastic net regularization. Lasso introduces a regularization term that penalizes high-value coefficients and, practically, removes highly correlated features from the model [192]. The main difference between Lasso and Ridge is that the latter does not penalize high-value coefficients that much and thus does not remove features from the model [193]. With regard to the elastic net regularization, this combines both Lasso and Ridge, inserting the corresponding penalties in the sum of square errors (i.e., SSE) loss function, thus addressing the issue of collinearity and performing feature selection [194].

In other ML problems, such as object recognition, speech synthesis and time series forecasting, regularization can also take place in the forms of data augmentation, early stopping, dropout and weight decay. Data augmentation is the process of artificially creating training data from the original data samples in order to enrich the training process of an algorithm [195]. Data augmentation is a quite common technique in DL [196], especially when it comes to text and image data [197]. The two main categories of regularization that can be found in the literature are data wrapping and oversampling, both achieving interesting results in specific use cases. However, it should be taken into consideration that the quality of the training and test data can inevitably affect the data augmentation phase; thus, any assumptions made for the distributions of the said data should be well justified [198]. Inappropriate data augmentation techniques can introduce bias in the ML lifecycle, but can also deflate bias coming from the dataset, as shown in numerous approaches such as the ones presented in [199–202].

With regard to early stopping, this is another popular regularization technique that is widely used in model training [203]. More specifically, early stopping refers to limiting the number of iterations (i.e., epochs), and training the model until it achieves the lowest possible training error before the validation error starts to increase. What is more, early stopping can improve the accuracy of ML algorithms and more specifically NNs by helping them to deal with noisy data, as discussed in [204].

Apart from early stopping, in NNs there also exist two additional strategies of regularization, namely dropout and weight decay. Regarding dropout, it randomly alters

the architecture of an NN in order to allow the model to avoid pure generalization to unseen data [205]. Even though it is a simple concept, it still requires the finetuning of hyperparameters, such as the learning rate and the number of units in the NN's hidden layer(s), so that it is indeed beneficial for the model [206].

As for the weight decay, it is quite similar to dropout; however, the latter penalizes the NN's complexity exponentially, whilst dropout penalizes the NN's complexity linearly [207]. Among the different weight decay methods that have been proposed in the literature, the most recent ones are Selective Weight Decay (i.e., SWD) [208] and Adaptive Weight Decay (AdaDecay) [209]. At this point, it should be mentioned that many sources in the literature conflict weight decay with L2 regression, while others clearly distinguish each other. This should be resolved in future scholarship, in order to avoid any confusion between those two concepts.

There exist several studies in the literature that utilize the aforementioned regularization techniques. In deeper detail, Lasso regression has been used alongside the RF algorithm for classifying long non-coding RNAs [210]. In [211], the authors compare early stopping with Bayesian regularization in an NN architecture that predicts the vapor pressure of pure ionic liquids, showcasing that the latter outperformed the first one in terms of generalization performance.

With regard to computer vision tasks, in [212], two versions of dropout are showcased, namely Biased and Crossmap dropouts. They both offer higher performance to the CNNs that utilize them. Similarly, in [213] weight decay, which is also used in [214], and dropout are also utilized in a CNN that performs facial emotion recognition, enabling it to achieve higher accuracy. In another computer vision task related to geological data, the authors utilize two types of data augmentation as a way of regularization in their developed CNN, namely CutOut and CutMix [215]. By doing so, they managed to minimize training loss and validation loss, thus achieving a performance equal to a transfer learning method that utilized a specific domain model. Moreover, in [216] the authors utilize a derivative of Lasso for an image classification task, and they prove that their approach is not only slightly inferior to other regularization techniques but also the corresponding NN needs fewer features to achieve those inferior results.

With regard to the energy domain, the authors in [217] use Lasso regularization in a linear regression model for time series forecasting while in another approach [218], a subtype of elastic net regularization named Adaptive Elastic Net is utilized in a multilevel regression model for estimating residential energy consumption. In [219], where the authors aim to predict rainfall runoff, they make use of early stopping and weight decay, thus simultaneously decreasing the model's overall training time whilst ensuring the highest accuracy of predictions possible.

As for the health domain, numerous approaches can be found in the literature that make use of different types of regularization techniques. More specifically, Lasso regularization is used in [220] alongside a regression model for subgroup identification of conditioning regimens in patients with Acute Myeloid Leukemia (i.e., AML) in Complete Remission (i.e., CR). A variation of Lasso is also used in [221], where the authors perform non-linear regression for tracking the position of the head and neck based on corresponding images. Elastic net is used in [222] and in [223]. In both cases, the authors make use of Magnetic Resonance Imaging (i.e., MRI) data alongside a logistic regression model and a linear regression model, respectively, achieving remarkable results in terms of classification accuracy. A variation of elastic net, named adjusted adaptive regularized logistic regression (i.e., AAELastic), is also utilized in [224] for cancer classification since it also allows the corresponding model to achieve the highest classification accuracy. With regard to dropout, this has been used in the CNN presented in [225], where the authors work on the brain tumor segmentation task. In a similar case [226], the authors utilize a combination of dropout and data augmentation in their CNN in order to estimate brain age based on MRI images, thus achieving state-of-the-art results. In another case regarding the diagnosis of

Parkinson's disease, the authors make use of a combination of dropout and L2 regression in a CNN-LSTM classifier again achieving promising results [227].

With regard to the traffic domain, Lasso regression has been adopted by various approaches. In [228], Lasso is being used for the evaluation of short and long-term effects of traffic policies, based on data coming from vehicles. In [229], the same regularization method is being used for estimating how crowded the means of public transport are, based on a set of different parameters, such as time, origin and destination stop. For predicting the severity of a crash, Lasso is also being utilized alongside a linear regression model, as shown in [230], thus providing promising results. Lasso is also being used in the context of network traffic since the authors in [231] prove that it provides a good trade-off between bias and variance when trying to estimate the day-to-day network demand. Lastly, in two more generic approaches, the authors utilize early stopping [232] and L2 regression [233], respectively. In the first case, the utilized NN makes use of the SDG optimizer, whose bias is being reduced due to early stopping, whilst in the second case the authors use an NN with a single layer, whose bias is also being reduced due to the application of L2 regression.

A complete list of studies regarding regularization techniques and that also take bias into consideration can be found in Table 10.

**Table 10.** Complete list of studies related to regularization techniques.

ID	Year	Ref.	Domain	Data Category	Regularization Technique
RS1	2019	[210]	Biology	Numeric	Lasso
RS2	2017	[211]	Chemistry	Numeric	Early stopping
RS3	2018	[212]	Computer Vision	Images	Dropout
RS4	2023	[213]	Computer Vision	Images	Weight Decay and dropout
RS5	2020	[214]	Computer Vision	Images	Weight Decay
RS6	2022	[215]	Computer Vision (Geology)	Images	Data augmentation
RS7	2017	[216]	Computer vision	Images	Lasso
RS8	2021	[217]	Energy	Time series Data	Lasso
RS9	2019	[218]	Energy	Numeric	Elastic Net
RS10	2012	[219]	Environment	Numeric	Early stopping and weight decay
RS11	2023	[220]	Health	Numeric	Lasso
RS12	2022	[221]	Health	Images	Lasso
RS13	2022	[222]	Health	Images	Elastic Net
RS14	2015	[223]	Health	Images	Elastic Net
RS15	2015	[224]	Health	Numeric	Elastic Net
RS16	2018	[225]	Health	Images	Dropout
RS17	2021	[226]	Health	Images	Data augmentation and dropout
RS18	2022	[227]	Health	Signals	L2 and Dropout
RS19	2020	[228]	Transport	Vehicle Data	Lasso
RS20	2020	[229]	Transport	Numeric	Lasso
RS21	2020	[230]	Transport	Numeric	Lasso and Elastic Net
RS22	2018	[231]	Networks	Numeric	Lasso
RS23	2020	[232]	Not Specific	Not Specific	Early stopping
RS24	2019	[233]	Not Specific	Numeric	L2

Based on the above table, it would be useful to compare the selected regularization techniques per approach and understand when each one of them is more appropriate to be

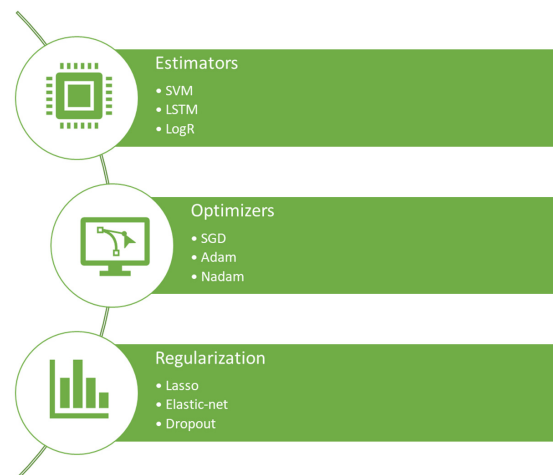
used, thus minimizing the bias introduced into the model. Generally, Lasso is used when the dataset consists of many features, since it is also able to perform feature selection, and when most coefficients are equal to zero. Ridge, on the other hand, is used when most of the coefficients are not equal to zero [234], while elastic net is a combination of them and can be used when the other two do not provide the appropriate results. Data augmentation is mostly suitable for image and text data where it is needed to increase the training samples. Dropout is more beneficial when it comes to training complex NN architectures on less complex datasets. Otherwise, weight decay is preferred [235]. Lastly, early stopping should usually be utilized regardless of the NN architecture, and the early stopping point should be set just before the model starts overfitting. The above-mentioned techniques are also summarized in Table 11.

**Table 11.** Recommended usage of Lasso, Ridge, Elastic Net, Data Augmentation, Dropout, Weight Decay.

Regularization Method	Recommended Usage
Lasso	Most feature coefficients are equal to zero
Ridge	Most feature coefficients are not equal to zero
Elastic Net	Need a combination of Lasso and Ridge
Data Augmentation	Image and text data with not sufficient samples
Dropout	Complex NN architectures and less complex data
Weight Decay	Dropout is not suitable
Early Stopping	Good practice as long as the threshold is set just before the model starts overfitting

#### 4.4. Common Mitigation Techniques for Algorithm Bias

According to the above-mentioned findings, there exist some common mitigation techniques that can be applied to address the bias that can be caused by ML algorithms. Those, as stated above, include adversarial debiasing, fairness constraints, variational autoencoders, contrastive learning and neural style transfer. By comparing the above-mentioned techniques, it is clear that fairness constraints are a set of restrictions that can be applied to ensure fairness when training a model, adversarial debiasing is based on training two models simultaneously, whilst variational autoencoders, contrastive learning and neural style transfer are an entirely different set of algorithms and architectures. All of them mainly focus on mitigating bias with respect to the existence of protected attributes in the datasets, such as gender or race. With regard to the comparative studies of ML algorithms, optimizers and regularization techniques that were previously presented, a summarization of the most common estimators, optimizers and regularization techniques that performed the best is depicted in Figure 4, as a summary of Section 4.



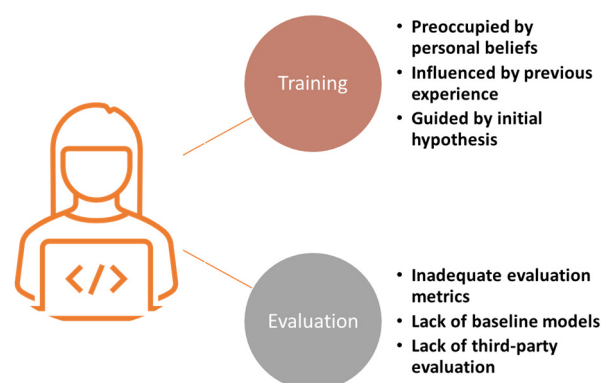
**Figure 4.** A summarization of the most common estimators, optimizers and regularization techniques.

## 5. Engineer Bias

The third factor that can affect ML models in terms of introducing bias is the ML engineers who are responsible for training and/or evaluating such models. ML model builders are prone to implicit bias, which occurs when the engineers make assumptions about the model that they are training based on their own experiences and personal beliefs that do not necessarily apply in general [236]. There exists a plethora of implicit biases in the literature, with the most identified one being when training and/or evaluating ML models, referring to confirmation bias. In confirmation bias, ML engineers unconsciously process the data and train the ML models in a way that will affirm their initial hypothesis or general beliefs [237].

Confirmation bias is a type of cognitive bias and, as such, there have been proposed several techniques that try to mitigate it. Those techniques are mainly theoretical frameworks that do not focus on ML engineers specifically but on analysts in general [238]. A plan for mitigating confirmation bias has been proposed in [239], where it is suggested that an analysis should be teamwork, and the analysts should exchange ideas prior to solving a problem. This, in the domain of ML model training, could be translated to discussing a possible solution to an ML problem with other ML engineers, who could possibly have different perspectives, thus leading to diverse approaches and techniques that could be applied to solve a specific problem, based on the perspectives of multiple engineers and not the perspective of just one. Another approach for reducing confirmation bias refers to “considering the opposite” [240]. In terms of ML, this means that the engineers would not keep on training a model until the results align with their initial hypothesis, thus avoiding the so-called experimenter’s bias [241].

ML engineers can also introduce bias during the model’s evaluation. More specifically, many proposed approaches in the literature do not apply adequate statistical methods for evaluating the proposed ML models [242]. Especially in NN architectures, k-fold testing (i.e., randomly partitioning original data sample in k equal-sized folds) is often being overlooked, even though it is well known that each different random initialization of the parameters of a model can drastically affect the model’s results [243]. Moreover, many approaches fail to compare the developed ML models to null (i.e., baseline) models. For instance, achieving 95% accuracy when classifying images of plants is trivial, while having the same accuracy for a model that predicts whether a person will develop cancer would be life-changing. On top of that, there is a lack of third-party evaluation and many NN architectures that are proposed in the literature cannot be verified [244]. Overall, ML engineers can introduce bias in their approaches due to the reasons that are depicted in Figure 5.

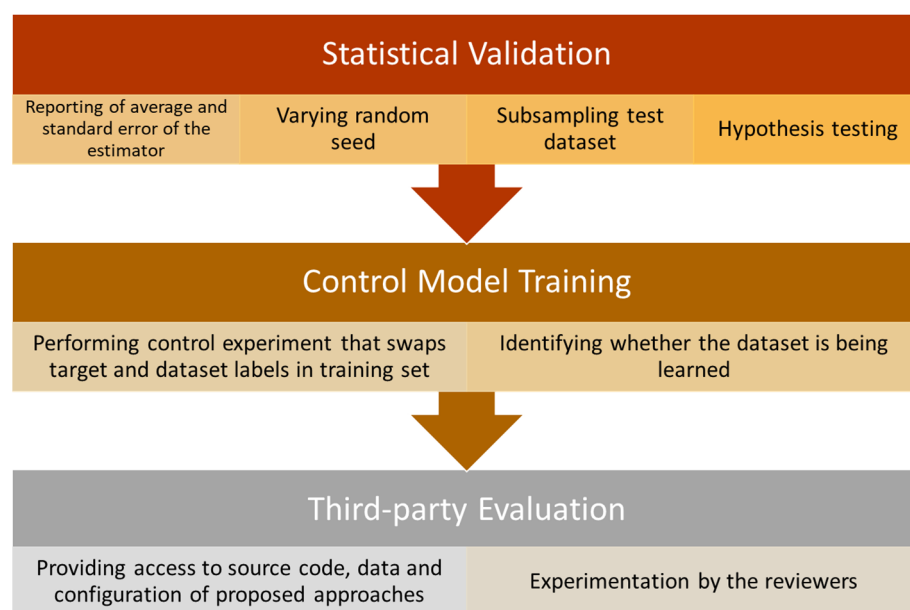


**Figure 5.** How can engineers introduce bias in the ML lifecycle?

To mitigate those issues in model evaluation, the authors in [242] propose a three-step action plan that consists of statistically validating results, training a dataset estimator as a



control model and conducting third-party evaluation of ML models. This action plan is summarized in Figure 6.



**Figure 6.** Action plan for reducing bias during the evaluation of ML models.

Following the above-mentioned methodology, the bias introduced by the ML engineers could be addressed to a great extent. In deeper detail, by experimenting with different values of random seed, performing hypothesis testing, subsampling the test data and monitoring and reporting the standard error of the trained estimator, would address the issue of the lack of statistical validation. Furthermore, the model training could be better controlled by observing how the model is affected by swapping targets and labels in the training dataset. With regard to the lack of third-party evaluation, this could be addressed by collaborating with other experienced engineers and providing access to source code, data and configuration, thus making the whole evaluation process more transparent and objective. To minimize the bias that could be caused by ML engineers, all the methods and techniques that were previously mentioned should be followed step by step as depicted in the above figure. They all aim to address the said bias during different stages of ML model training and evaluation, being complementary to each other.

## 6. Discussion

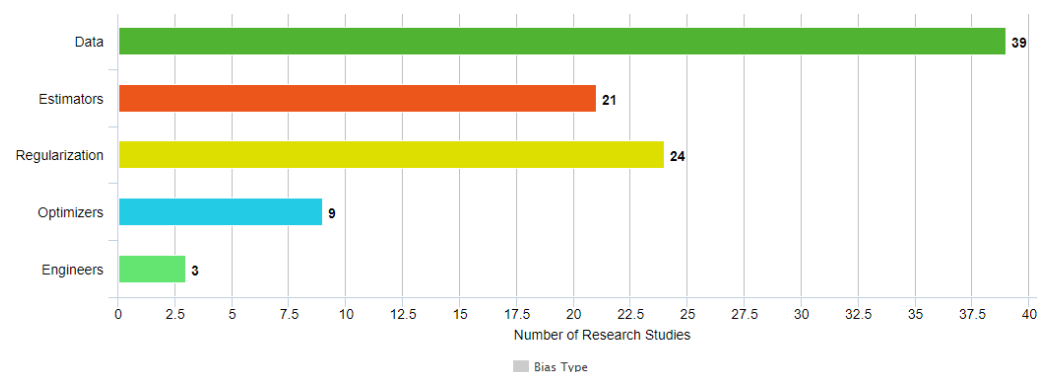
Bias, whether it originates from humans or the underlying mathematics of ML models, is a key factor in ML that actively affects the results of such models. The research community is aware of this influence, which is why several approaches have been proposed to try and mitigate it, as described in the previous sections. What is more, there also exist some initiatives, such as the AI Fairness 360 [245], Fairlearn [246] and Fairkit-learn [247] tools that aim to provide a more complete technical framework for mitigating bias when training ML models, thus promoting AI fairness. These kinds of frameworks have showcased promising results in terms of bias mitigation [248]; however, they still have a lot of room for improvement, which is why their developers request the active engagement of researchers so that they can be further enhanced and improved [249].

Removing all types of bias entirely from the ML lifecycle is not an easy task and some would argue that it is impossible. The first and foremost concern when training and/or evaluating ML models is identifying the existence of bias and the corresponding types of bias that might occur in terms of the provided data, the models themselves, as well as the bias that can be introduced by ML engineers, unconsciously or not. This identification can be conducted either empirically by taking into consideration other similar approaches

and use cases or by utilizing algorithms that are capable of identifying bias, as the ones presented in this manuscript. Then, the corresponding countermeasures should be taken into consideration and applied in order to minimize the identified biases. To this end, this manuscript's scope is to serve as a reference point for ML engineers and everyone who is interested in developing, evaluating and/or utilizing ML models, which they can consult so that they can firstly identify bias and then debias their approaches.

Based on the literature review that was undertaken, it appears that most of the current approaches mainly focus on the existence of bias in the training datasets (i.e., data bias) and claim to have applied some kind of corresponding bias mitigation technique, like the IPW adjustment technique, the Imitate algorithm and its derivatives. With regard to the bias caused by the ML models and especially the selection of the appropriate optimizer and regularization method, little attention is paid to this matter, whilst most of the proposed approaches do not take into consideration that the choice of an optimizer and regularization method can also introduce bias in the model, as mentioned in Section 4.2.

Overall, from all the research documents that were analyzed in the context of this manuscript, 39 papers analyzed methods and techniques for identifying and mitigating data bias, 21 comparative studies highlighted the importance of selecting the proper estimator to avoid the introduction of further bias and 24 comparative studies analyzed the way that a regularization technique may assist in reducing algorithmic bias. With regard to the way that an optimizer can affect a model in terms of bias and how an ML engineer can introduce bias, only nine and three studies were identified in the literature, respectively, as shown in Figure 7.



**Figure 7.** Number of research studies per bias type.

The latter signifies that future research approaches should aim to (i) better justify the selection of a specific optimizer and how it may minimize bias in contrast to other available optimizers and (ii) evaluate their proposed approaches based on the appropriate metrics and offer transparency to their trained models by allowing the evaluation from third-party evaluators and reviewers.

As for other literature reviews that talk about bias in machine learning, they mostly focus on data bias as well. More specifically, in [250] the authors mainly focus on selection bias and bias caused by imbalanced data, whilst they present the most common techniques to address these types of biases. Similarly, in [251] the authors provide a comprehensive review with regard to methods that can mitigate human bias (e.g., selection bias) that is present in the data. Another interesting comprehensive review is provided by the authors in [252], where they have gathered a noticeable amount of research studies to identify types of bias and methods to mitigate them, focusing on data bias and estimator bias. However, the above-mentioned review does not take into consideration either the fact that model builders may also introduce bias when training/evaluating their models (i.e., engineer bias), as depicted in Section 5 of this manuscript, nor the fact that the selection of an optimizer may also affect an ML model in terms of bias, as stated in Section 4. A similar comprehensive review that focuses on the same aspects as the last-mentioned review is

presented in [253]. What is more, in contrast to this manuscript, the above-mentioned reviews do not categorize the provided approaches per use case domain, making it difficult to identify the most appropriate approach for a specific domain.

With regard to reviews that are more use-case-specific, those mostly refer to the health domain. More specifically, the authors in [254] provide a comprehensive study of the types of bias that can be encountered in health-related ML models and, more precisely, in ML models that assess the risk of cardiovascular disease (i.e., CVD). In this study, the authors also focused on data and estimator bias. In [255], the authors also investigate data bias in the context of CVD and provide ways to mitigate the bias that may be present in the electronic health records (i.e., EHRs) of patients. Similarly, the authors in [256] also provide a comprehensive list of approaches that tackle bias in radiology; however, they also refer to optimizers and regularization techniques, since they also identify that those can also introduce further bias. However, also in this case little to no attention is paid to the bias that can be introduced by the ML engineers. A summary of the above can be found in Table 12.

**Table 12.** Comparison of reviews related to bias mitigation.

ID	Year	Ref.	Data Bias	Estimator Bias	Optimizer Bias	Regularization Bias	Engineer Bias
LR1	2019	[250]	✓				
LR2	2018	[251]	✓				
LR3	2024	[252]	✓	✓			
LR4	2023	[253]	✓	✓			
LR5	2022	[254]	✓	✓			
LR6	2023	[255]	✓				
LR7	2022	[256]	✓	✓	✓	✓	
Proposed Approach			✓	✓	✓	✓	✓

The above provides a direction for future research related to ML models towards improving methodological practices to incorporate the necessary bias mitigation techniques, thus providing more complete and meaningful results. Having said that, the findings of this manuscript regarding algorithm bias show that the SVM algorithm is preferred to reduce bias whilst the SGD optimizer and the regularization method Lasso are the most widely used optimizer and regularization method, respectively. However, it should be highlighted that the above does not suggest that SVM, SGD and Lasso always produce the least bias. As it has already been stated multiple times in the manuscript, the type of data and the corresponding ML task that must be undertaken will dictate the most appropriate algorithm, optimizer and regularization method in order to reduce bias. Having this in mind, it is also quite useful to have access to other similar approaches and the methods that they use so that the model builders have some guidance regarding what techniques seem to be more appropriate for their use case. As for the model's evaluation, there also seems to be a lack of reference to the adaptation of appropriate techniques for mitigating bias and ensuring objectivity during the evaluation process. Based on the findings of this study, the three-step action plan presented in Figure 6 should become the cornerstone of the evaluation of ML models and related publications, thus leading to a widely adopted framework for achieving objective evaluation of ML approaches.

With regard to any limitations that this literature review may have, those are mainly related to potential selection bias when performing the selection process of the corresponding studies. The most representative search keywords were selected and used for each type of bias; however, the search was carried out based on the presence of these keywords on the title, abstract and the keywords of the corresponding manuscripts. This means that

there might exist publications that refer to the concepts that are analyzed throughout this manuscript in the main text of the publications. In order to address the aforementioned possibility, the search queries could be further updated to also search the full text of the publications, although this is a capability that is not offered by every publication database. Moreover, all the publications that are taken into consideration are written in English. Based on the search results, there exist a few publications that are written in other languages. These publications, which are mostly written in Chinese, were not taken into consideration. This limitation can be addressed by communicating with scientific personnel that know these foreign languages, in order to help with translating the publications to English. What is more, as already stated, this research provides a literature review regarding the concept of bias in AI and how to identify/mitigate it. As a future direction of this research, a framework could be implemented that integrates several techniques and algorithms for bias identification/mitigation that are analyzed in this manuscript to further assist AI practitioners and enthusiasts in identifying and mitigating bias in their approaches.

## 7. Conclusions

To summarize, bias is a key factor in ML that will further trouble researchers during the exponential growth and adaptation of AI systems in everyday life. It is imperative for everyone who is involved in training, evaluation or even using AI systems to (i) identify the biases that may be present in the underlying ML model and (ii) apply the appropriate methods to mitigate them. As mentioned earlier, bias is almost impossible to eliminate, but a more realistic goal is to minimize it. As shown in this manuscript, bias has many forms that can be identified during the different stages of the ML lifecycle. In the context of this manuscript, the types of bias were grouped into three main categories, namely bias originating from the data, the ML models and the ML engineers, respectively. For each of those categories, the types of bias were presented, as well as the methods and approaches that have been proposed in the literature in order to mitigate them. Based on the findings of this study, it appears that the current literature mainly focuses on specific biases in ML, mostly related to the data, whilst it either underestimates or omits other types of biases, especially when it comes to evaluating the proposed ML models.

Researchers undoubtedly try to mitigate bias in their approaches; however, they should start exploring other aspects of it and try to mitigate it throughout the whole ML lifecycle and not on isolated stages. To this end, this manuscript not only serves as a guideline for ML engineers regarding possible biases and ways to mitigate them but also provides a direction for future research, especially focusing on the existing terms and approaches of ML models' biases, where the ML engineers should better concentrate on. Of course, this manuscript could be further extended in terms of the existing AI fairness tools that, in the context of this study, were briefly described. Moreover, the context of this manuscript could be transformed into a technical implementation that would pave the way for a more complete mitigation of bias in ML, regardless of the provided use case scenario. The aforementioned technical implementation could then also be evaluated in the context of several related EU projects in which the authors of this manuscript have participated [257–262] to extract more valuable results and potentially update the technical implementation accordingly to better tackle bias-related challenges in the corresponding use cases.

**Author Contributions:** Conceptualization, K.M.; methodology, K.M., A.M. (Argyro Mavrogiorgou) and A.K.; validation, A.M. (Argyro Mavrogiorgou), A.K. and K.M.; formal analysis, K.M.; investigation, K.M. and A.M. (Andreas Menychtas); resources, K.M. and A.K.; writing—original draft preparation, K.M. and A.M. (Argyro Mavrogiorgou); writing—review and editing, K.M., A.M. (Argyro Mavrogiorgou), A.K. and A.M. (Andreas Menychtas); visualization, K.M.; supervision, A.M. (Andreas Menychtas) and D.K.; project administration, D.K.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research leading to the results presented in this paper has received funding from the European Union’s funded Projects AI4Gov under grant agreement no 101094905 and XR5.0 under grant agreement no 101135209.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Fazelpour, S.; Danks, D. Algorithmic bias: Senses, sources, solutions. *Philos. Compass* **2021**, *16*, e12760. [CrossRef]
2. Delgado-Rodriguez, M.; Llorca, J. Bias. *J. Epidemiol. Community Health* **2004**, *58*, 635–641. [CrossRef]
3. Statista—“Market Size and Revenue Comparison for Artificial Intelligence Worldwide from 2018 to 2030”. Available online: <https://www.statista.com/statistics/941835/artificial-intelligence-market-size-revenue-comparisons> (accessed on 15 February 2024).
4. Statista—“Share of Adults in the United States Who Were Concerned about Issues Related to Artificial Intelligence (AI) as of February 2023”. Available online: <https://www.statista.com/statistics/1378220/us-adults-concerns-about-artificial-intelligence-related-issues> (accessed on 15 February 2024).
5. Ray, P.P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 121–154. [CrossRef]
6. Meyer, J.G.; Urbanowicz, R.J.; Martin, P.C.N.; O’connor, K.; Li, R.; Peng, P.-C.; Bright, T.J.; Tatonetti, N.; Won, K.J.; Gonzalez-Hernandez, G.; et al. ChatGPT and large language models in academia: Opportunities and challenges. *BioData Min.* **2023**, *16*, 20. [CrossRef] [PubMed]
7. Yee, K.; Tantipongpipat, U.; Mishra, S. Image cropping on twitter: Fairness metrics, their limitations, and the importance of representation, design, and agency. In Proceedings of the ACM on Human-Computer Interaction, 5(CSCW2), Virtual, 23 October 2021; pp. 1–24.
8. Birhane, A.; Prabhu, V.U.; Whaley, J. Auditing saliency cropping algorithms. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5 January 2022; pp. 4051–4059.
9. Dressel, J.J. Accuracy and Racial Biases of Recidivism Prediction Instruments. Undergraduate Thesis, Dartmouth College, Hanover, NH, USA, 2017.
10. Lin, Z.; Jung, J.; Goel, S.; Skeem, J. The limits of human predictions of recidivism. *Sci. Adv.* **2020**, *6*, eaaz0652. [CrossRef] [PubMed]
11. Engel, C.; Linhardt, L.; Schubert, M. Code is law: How COMPAS affects the way the judiciary handles the risk of recidivism. *Artif. Intell. Law* **2024**, *32*, 1–22. [CrossRef]
12. Roselli, D.; Matthews, J.; Talagala, N. Managing bias in AI. In Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, 13 May 2019; pp. 539–544.
13. Kordzadeh, N.; Ghasemaghahi, M. Algorithmic bias: Review, synthesis, and future research directions. *Eur. J. Inf. Syst.* **2022**, *31*, 388–409. [CrossRef]
14. Schelter, S.; Stoyanovich, J. Taming technical bias in machine learning pipelines. *Bull. Tech. Comm. Data Eng.* **2020**, *43*, 39–50.
15. Ha, T.; Kim, S. Improving Trust in AI with Mitigating Confirmation Bias: Effects of Explanation Type and Debiasing Strategy for Decision-Making with Explainable AI. *Int. J. Hum.-Comput. Interact.* **2023**, *39*, 1–12. [CrossRef]
16. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P. Handling imbalanced datasets: A review. *GESTS Int. Trans. Comput. Sci. Eng.* **2006**, *30*, 25–36.
17. Yen, S.; Lee, Y. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. *Lect. Notes Control Inf. Sci.* **2006**, *344*, 731.
18. Yen, S.-J.; Lee, Y.-S. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Syst. Appl.* **2009**, *36*, 5718–5727. [CrossRef]
19. Tahir, M.A.; Kittler, J.; Yan, F. Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognit.* **2012**, *45*, 3738–3750. [CrossRef]
20. Elhassan, T.; Aljurf, M. Classification of imbalance data using totem link (t-link) combined with random under-sampling (rus) as a data reduction method. *Glob. J. Technol. Opt. S* **2016**, *1*, 100011. [CrossRef]
21. Fernandez, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
22. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Netw, (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1 June 2008; pp. 1322–1328.
23. Yang, X.; Kuang, Q.; Zhang, W.; Zhang, G. AMDO: An over-sampling technique for multi-class imbalanced problems. *IEEE Trans. Knowl. Data Eng.* **2017**, *30*, 1672–1685. [CrossRef]



24. Azaria, A. ChatGPT: More Human-Like Than Computer-Like, but Not Necessarily in a Good Way. In Proceedings of the 2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI), Atlanta, GA, USA, 6 November 2023; pp. 468–473.
25. Atreides, K.; Kelley, D. Cognitive Biases in Natural Language: Automatically Detecting, Differentiating, and Measuring Bias in Text. *Differentiating, and Measuring Bias in Text*. 2023. Available online: [https://www.researchgate.net/profile/Kyrtin-Atreides/publication/372078491\\_Cognitive\\_Biases\\_in\\_Natural\\_Language\\_Automatically\\_Detecting\\_Differentiating\\_and\\_Measuring\\_Bias\\_in\\_Text/links/64a3e11195bbbe0c6e0f149c/Cognitive-Biases-in-Natural-Language-Automatically-Detecting-Differentiating-and-Measuring-Bias-in-Text.pdf](https://www.researchgate.net/profile/Kyrtin-Atreides/publication/372078491_Cognitive_Biases_in_Natural_Language_Automatically_Detecting_Differentiating_and_Measuring_Bias_in_Text/links/64a3e11195bbbe0c6e0f149c/Cognitive-Biases-in-Natural-Language-Automatically-Detecting-Differentiating-and-Measuring-Bias-in-Text.pdf) (accessed on 15 February 2024).
26. Blawatt, K.R. Appendix A: List of cognitive biases. In *Marconomics*; Emerald Group Publishing Limited: Bingley, UK, 2016; pp. 325–336.
27. Sayão, L.F.; Baião, F.A. An Ontology-based Data-driven Architecture for Analyzing Cognitive Biases in Decision-making. In Proceedings of the XVI Seminar on Ontology Research in Brazil (ONTOBRAS 2023) and VII Doctoral and Masters Consortium on Ontologies, (WTDO 2023), Brasilia, Brazil, 28 August–1 September 2023.
28. Harris, G. Methods to Evaluate Temporal Cognitive Biases in Machine Learning Prediction Models. In Proceedings of the Companion Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 572–575.
29. Liu, Q.; Jiang, H.; Pan, Z.; Han, Q.; Peng, Z.; Li, Q. BiasEye: A Bias-Aware Real-time Interactive Material Screening System for Impartial Candidate Assessment. In Proceedings of the IUI '24: 29th International Conference on Intelligent User Interfaces, Greenville, SC, USA, 18–21 March 2024; pp. 325–343.
30. Harris, G. Mitigating cognitive biases in machine learning algorithms for decision making. In Proceedings of the Companion Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 775–781.
31. Chen, X.; Sun, R.; Saluz, U.; Schiavon, S.; Geyer, P. Using causal inference to avoid fallouts in data-driven parametric analysis: A case study in the architecture, engineering, and construction industry. *Dev. Built Environ.* **2023**, *17*, 100296. [[CrossRef](#)]
32. Kavitha, J.; Kiran, J.; Prasad, S.D.V.; Soma, K.; Babu, G.C.; Sivakumar, S. Prediction and Its Impact on Its Attributes While Biasing Machine Learning Training Data. In Proceedings of the 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 16 December 2022; pp. 1–7.
33. Schmidgall, S. Addressing cognitive bias in medical language models. *arXiv* **2024**, arXiv:2402.08113.
34. Bareinboim, E.; Tian, J.; Pearl, J. Recovering from selection bias in causal and statistical inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 433–450.
35. Tripepi, G.; Jager, K.J.; Dekker, F.W.; Zoccali, C. Selection bias and information bias in clinical research. *Nephron Clin. Pract.* **2010**, *115*, c94–c99. [[CrossRef](#)]
36. Smith, L.H. Selection mechanisms and their consequences: Understanding and addressing selection bias. *Curr. Epidemiol. Rep.* **2020**, *7*, 179–189. [[CrossRef](#)]
37. Mendez, M.; Maathuis, B.; Hein-Griggs, D.; Alvarado-Gamboa, L.-F. Performance evaluation of bias correction methods for climate change monthly precipitation projections over costa rica. *Water* **2020**, *12*, 482. [[CrossRef](#)]
38. Heo, J.-H.; Ahn, H.; Shin, J.-Y.; Kjeldsen, T.R.; Jeong, C. Probability distributions for a quantile mapping technique for a bias correction of precipitation data: A case study to precipitation data under climate change. *Water* **2019**, *11*, 1475. [[CrossRef](#)]
39. Soriano, E.; Mediero, L.; Garijo, C. Selection of bias correction methods to assess the impact of climate change on flood frequency curves. *Water* **2019**, *11*, 2266. [[CrossRef](#)]
40. Kaltenpoth, D.; Vreeken, J. Identifying selection bias from observational data. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7 February 2023; Volume 37, pp. 8177–8185.
41. Gharib, A.; Davies, E.G. A workflow to address pitfalls and challenges in applying machine learning models to hydrology. *Adv. Water Resour.* **2021**, *152*, 103920. [[CrossRef](#)]
42. Shen, Z.; Cui, P.; Kuang, K.; Li, B.; Chen, P. Causally regularized learning with agnostic data selection bias. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22 October 2018; pp. 411–419.
43. Bibi, S.; Shin, J. Detection of Face Features using Adapted Triplet Loss with Biased data. In Proceedings of the 2022 IEEE International Conference on Imaging Systems and Techniques (IST), Virtual, 21 June 2022; pp. 1–6.
44. Yang, Q.; Chen, Z.; Yuan, Y. Hierarchical bias mitigation for semi-supervised medical image classification. *IEEE Trans. Med. Imaging* **2023**, *42*, 2200–2210. [[CrossRef](#)] [[PubMed](#)]
45. Wu, P.; Xu, T.; Wang, Y. Learning Personalized Treatment Rules from Electronic Health Records Using Topic Modeling Feature Extraction. In Proceedings of the 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Washington, DC, USA, 5 October 2019; pp. 392–402.
46. Samadani, A.; Wang, T.; van Zon, K.; Celi, L.A. VAP risk index: Early prediction and hospital phenotyping of ventilator-associated pneumonia using machine learning. *Artif. Intell. Med.* **2023**, *146*, 102715. [[CrossRef](#)] [[PubMed](#)]
47. Wang, H.; Kuang, K.; Lan, L.; Wang, Z.; Huang, W.; Wu, F.; Yang, W. Out-of-distribution generalization with causal feature separation. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 1758–1772. [[CrossRef](#)]
48. Yang, Z.; Liu, Y.; Ouyang, C.; Ren, L.; Wen, W. Counterfactual can be strong in medical question and answering. *Inf. Process. Manag.* **2023**, *60*, 103408. [[CrossRef](#)]
49. Costello, M.J.; Li, Y.; Zhu, Y.; Walji, A.; Sousa, S.; Remers, S.; Chorny, Y.; Rush, B.; MacKillop, J. Using conventional and machine learning propensity score methods to examine the effectiveness of 12-step group involvement following inpatient addiction treatment. *Drug Alcohol. Depend.* **2021**, *227*, 108943. [[CrossRef](#)]



50. Liu, X.; Ai, W.; Li, H.; Tang, J.; Huang, G.; Feng, F.; Mei, Q. Deriving user preferences of mobile apps from their management activities. *ACM Trans. Inf. Syst.* **2017**, *35*, 1–32. [CrossRef]
51. Minatel, D.; Parmezan, A.R.; Cúri, M.; Lopes, A.D.A. Fairness-Aware Model Selection Using Differential Item Functioning. In Proceedings of the 2023 International Conference on Machine Learning and Applications (ICMLA), Jacksonville, FL, USA, 15 December 2023; pp. 1971–1978.
52. Dost, K.; Taskova, K.; Riddle, P.; Wicker, J. Your best guess when you know nothing: Identification and mitigation of selection bias. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17 November 2020; pp. 996–1001.
53. GitHub—Imitate. Available online: <https://github.com/KatDost/Imitate> (accessed on 15 February 2024).
54. Dost, K.; Duncanson, H.; Ziogas, I.; Riddle, P.; Wicker, J. Divide and imitate: Multi-cluster identification and mitigation of selection bias. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Chengdu, China, 11 May 2022; pp. 149–160.
55. Shi, L.; Li, S.; Ding, X.; Bu, Z. Selection bias mitigation in recommender system using uninteresting items based on temporal visibility. *Expert Syst. Appl.* **2023**, *213*, 118932. [CrossRef]
56. Liu, H. Rating distribution calibration for selection bias mitigation in recommendations. In Proceedings of the ACM Web Conference, Lyon, France, 25 April 2022; pp. 2048–2057.
57. Liu, F.; Cole, J.; Eisenschlos, J.M.; Collier, N. Are Ever Larger Octopi Still Influenced by Reporting Biases? 2022. Available online: <https://research.google/pubs/are-ever-larger-octopi-still-influenced-by-reporting-biases/> (accessed on 15 February 2024).
58. Shwartz, V.; Choi, Y. Do neural language models overcome reporting bias? In Proceedings of the 28th International Conference on Computational Linguistics, Virtual, 8 December 2020; pp. 6863–6870.
59. Cai, B.; Ding, X.; Chen, B.; Du, L.; Liu, T. Mitigating Reporting Bias in Semi-supervised Temporal Commonsense Inference with Probabilistic Soft Logic. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 10454–10462. [CrossRef]
60. Wu, Q.; Zhao, M.; He, Y.; Huang, L.; Ono, J.; Wakaki, H.; Mitsufuji, Y. Towards reporting bias in visual-language datasets: Bimodal augmentation by decoupling object-attribute association. *arXiv* **2023**, arXiv:2310.01330.
61. Chiou, M.J.; Ding, H.; Yan, H.; Wang, C.; Zimmermann, R.; Feng, J. Recovering the unbiased scene graphs from the biased ones. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 17 October 2021; pp. 1581–1590.
62. Misra, I.; Lawrence Zitnick, C.; Mitchell, M.; Girshick, R. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June 2016; pp. 2930–2939.
63. Atay, M.; Gipson, H.; Gwyn, T.; Roy, K. Evaluation of gender bias in facial recognition with traditional machine learning algorithms. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5 December 2021; pp. 1–7.
64. Ayoade, G.; Chandra, S.; Khan, L.; Hamlen, K.; Thuraisingham, B. Automated threat report classification over multi-source data. In Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Vancouver, BC, Canada, 12 November 2018; pp. 236–245.
65. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Venkatraman, S. Robust intelligent malware detection using deep learning. *IEEE Access* **2019**, *7*, 46717–46738. [CrossRef]
66. Hinchliffe, C.; Rehman, R.Z.U.; Branco, D.; Jackson, D.; Ahmaniemi, T.; Guerreiro, T.; Chatterjee, M.; Manyakov, N.V.; Pandis, I.; Davies, K.; et al. Identification of Fatigue and Sleepiness in Immune and Neurodegenerative Disorders from Measures of Real-World Gait Variability. In Proceedings of the 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Sydney, Australia, 24 July 2023; pp. 1–4.
67. Bughin, J.; Cincera, M.; Peters, K.; Reykowska, D.; Żyszkiewicz, M.; Ohme, R. Make it or break it: On-time vaccination intent at the time of Covid-19. *Vaccine* **2023**, *41*, 2063–2072. [CrossRef] [PubMed]
68. Seo, D.-C.; Han, D.-H.; Lee, S. Predicting opioid misuse at the population level is different from identifying opioid misuse in individual patients. *Prev. Med.* **2020**, *131*, 105969. [CrossRef]
69. Catania, B.; Guerrini, G.; Janpoh, Z. Mitigating Representation Bias in Data Transformations: A Constraint-based Optimization Approach. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15 December 2023; pp. 4127–4136.
70. Hu, Q.; Rangwala, H. Metric-free individual fairness with cooperative contextual bandits. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17 November 2020; pp. 182–191.
71. Rengasamy, D.; Mase, J.M.; Rothwell, B.; Figueredo, G.P. An intelligent toolkit for benchmarking data-driven aerospace prognostics. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27 October 2019; pp. 4210–4215.
72. Bao, F.; Deng, Y.; Zhao, Y.; Suo, J.; Dai, Q. Bosco: Boosting corrections for genome-wide association studies with imbalanced samples. *IEEE Trans. NanoBiosci.* **2017**, *16*, 69–77. [CrossRef] [PubMed]
73. Tiwari, V.; Verma, M. Prediction Of Groundwater Level Using Advance Machine Learning Techniques. In Proceedings of the 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubali, India, 23 June 2023; pp. 1–6.
74. Behfar, S.K. Decentralized intelligence and big data analytics reciprocal relationship. In Proceedings of the 2023 Fifth International Conference on Blockchain Computing and Applications (BCCA), Bristol, UK, 13 November 2023; pp. 643–651.

75. Sepasi, S.; Etemadi, H.; Pasandidehfard, F. Designing a Model for Financial Reporting Bias. *J. Account. Adv.* **2021**, *13*, 161–189.
76. Al-Sarraj, W.F.; Lubbad, H.M. Bias detection of Palestinian/Israeli conflict in western media: A sentiment analysis experimental study. In Proceedings of the 2018 International Conference on Promising Electronic Technologies (ICPET), Hyderabad, India, 28 December 2018; pp. 98–103.
77. Shumway, R.H.; Stoffer, D.S.; Shumway, R.H.; Stoffer, D.S. ARIMA Models. Time Series Analysis and Its Applications: With R Examples. 2017; pp. 75–163. Available online: <https://link.springer.com/book/9783031705830> (accessed on 24 February 2024).
78. Salleh, M.N.M.; Talpur, N.; Hussain, K. Adaptive neuro-fuzzy inference system: Overview, strengths, limitations, and solutions. In Proceedings of the Data Mining and Big Data: Second International Conference, Fukuoka, Japan, 27 July 2017; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 527–535.
79. Teodorović, D. Bee colony optimization (BCO). In *Innovations in Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2009.
80. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9 December 2019; pp. 3285–3292.
81. Kramer, O. Cascade support vector machines with dimensionality reduction. In *Applied Computational Intelligence and Soft Computing*; Wiley Online Library: Hoboken, NJ, USA, 2015; p. 216132.
82. Ruggieri, S. Efficient C4. 5 [classification algorithm]. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 438–444. [[CrossRef](#)]
83. Lewis, R.J. An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine*; Department of Emergency Medicine Harbor-UCLA Medical Center Torrance: San Francisco, CA, USA, 2002.
84. Lu, W.; Li, J.; Wang, J.; Qin, L. A CNN-BiLSTM-AM method for stock price prediction. *Neural Comput. Appl.* **2021**, *33*, 4741–4753. [[CrossRef](#)]
85. Wallach, H.M. *Conditional Random Fields: An Introduction*; CIS: East Greenbush, NY, USA, 2004.
86. Mustaqeem, K.S. CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network. *Mathematics* **2020**, *8*, 2133. [[CrossRef](#)]
87. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [[CrossRef](#)] [[PubMed](#)]
88. Kumar, D.; Klefsjö, B. Proportional hazards model: A review. *Reliab. Eng. Syst. Saf.* **1994**, *44*, 177–188. [[CrossRef](#)]
89. Kuhn, M.; Weston, S.; Keefer, C.; Coulter, N. Cubist Models for Regression. R Package Vignette R Package Version 0.0. 2012; 18; 480. Available online: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=fd880d2b4482fc9b383435d51f6d730c02e0be36> (accessed on 20 February 2024).
90. Song, Y.Y.; Ying, L.U. Decision tree methods: Applications for classification and prediction. *Shanghai Archiv. Psychiatry* **2015**, *27*, 130.
91. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**, arXiv:1605.07678.
92. Brim, A. Deep reinforcement learning pairs trading with a double deep Q-network. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6 January 2020; pp. 0222–0227.
93. Gardner, E.S., Jr. Exponential smoothing: The state of the art—Part II. *Int. J. Forecast.* **2006**, *22*, 637–666. [[CrossRef](#)]
94. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [[CrossRef](#)]
95. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13 August 2016; pp. 785–794.
96. Zhong, J.; Feng, L.; Ong, Y.S. Gene expression programming: A survey. *IEEE Comput. Intell. Mag.* **2017**, *12*, 54–72. [[CrossRef](#)]
97. Prettenhofer, P.; Louppe, G. Gradient boosted regression trees in scikit-learn. In Proceedings of the PyData, London, UK, 21–23 February 2014.
98. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*; IEEE: Boston, MA, USA, 2017; pp. 1597–1600.
99. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In Proceedings of the On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Italy, 3 November 2003; pp. 986–996.
100. Fan, J.; Ma, X.; Wu, L.; Zhang, F.; Yu, X.; Zeng, W. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. *Agric. Water Manag.* **2019**, *225*, 105758. [[CrossRef](#)]
101. Santoso, N.; Wibowo, W. Financial distress prediction using linear discriminant analysis and support vector machine. *J. Phys. Conf. Ser.* **2019**, *979*, 012089. [[CrossRef](#)]
102. Su, X.; Yan, X.; Tsai, C.L. Linear regression. *Wiley Interdiscip. Rev. Comput. Stat.* **2012**, *4*, 275–294. [[CrossRef](#)]
103. Joachims, T. Training linear SVMs in linear time. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20 August 2006; pp. 217–226.
104. Connelly, L. Logistic regression. *Medsurg Nurs.* **2020**, *29*, 353–354.
105. Sherstinsky, A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
106. Kruse, R.; Mostaghim, S.; Borgelt, C.; Braune, C.; Steinbrecher, M. Multi-layer perceptrons. In *Computational Intelligence: A Methodological Introduction*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 53–124.

107. Abbas, M.; Memon, K.A.; Jamali, A.A.; Memon, S.; Ahmed, A. Multinomial Naive Bayes classification model for sentiment analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 62.
108. Wu, Y.-C.; Feng, J.-W. Development and application of artificial neural network. *Wirel. Pers. Commun.* **2018**, *102*, 1645–1656. [[CrossRef](#)]
109. Rigatti, S.J. Random forest. *J. Insur. Med.* **2017**, *47*, 31–39. [[CrossRef](#)]
110. Joslin, D.E.; Clements, D.P. Squeaky wheel optimization. *J. Artif. Intell. Res.* **1999**, *10*, 353–373. [[CrossRef](#)]
111. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E.; Terpenning, I. STL: A seasonal-trend decomposition. *J. Off. Stat.* **1990**, *6*, 3–73.
112. Wang, H.; Hu, D. Comparison of SVM and LS-SVM for regression. In Proceedings of the 2005 International Conference on Neural Netw. and Brain, Beijing, China, 13–15 October 2005; Volume 1, pp. 279–283.
113. Li, X.; Lv, Z.; Wang, S.; Wei, Z.; Wu, L. A reinforcement learning model based on temporal difference algorithm. *IEEE Access* **2019**, *7*, 121922–121930. [[CrossRef](#)]
114. Ramos, J. Using tf-idf to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning, Los Angeles, CA, USA, 23 June 2003; Volume 242, pp. 29–48.
115. Cicirello, V.A.; Smith, S.F. Enhancing stochastic search performance by value-biased randomization of heuristics. *J. Heuristics* **2005**, *11*, 5–34. [[CrossRef](#)]
116. Stock, J.H.; Watson, M.W. Vector autoregressions. *J. Econ. Perspect.* **2001**, *15*, 101–115. [[CrossRef](#)]
117. Biney, J.K.M.; Vašát, R.; Bell, S.M.; Kebonye, N.M.; Klement, A.; John, K.; Borůvka, L. Prediction of topsoil organic carbon content with Sentinel-2 imagery and spectroscopic measurements under different conditions using an ensemble model approach with multiple pre-treatment combinations. *Soil Tillage Res.* **2022**, *220*, 105379. [[CrossRef](#)]
118. Lihu, A.; Holban, S. Top five most promising algorithms in scheduling. In Proceedings of the 2009 5th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, 28 May 2009; pp. 397–404.
119. Wu, S.G.; Wang, Y.; Jiang, W.; Oyetunde, T.; Yao, R.; Zhang, X.; Shimizu, K.; Tang, Y.J.; Bao, F.S. Rapid Prediction of Bacterial Heterotrophic Fluxomics Using Machine Learning and Constraint Programming. *PLoS Comput. Biol.* **2016**, *12*, e1004838. [[CrossRef](#)] [[PubMed](#)]
120. Rafay, A.; Suleman, M.; Alim, A. Robust review rating prediction model based on machine and deep learning: Yelp dataset. In Proceedings of the 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), Karachi, Pakistan, 26 March 2020; pp. 8138–8143.
121. Wescoat, E.; Kerner, S.; Mears, L. A comparative study of different algorithms using contrived failure data to detect robot anomalies. *Procedia Comput. Sci.* **2022**, *200*, 669–678. [[CrossRef](#)]
122. Velasco-Gallego, C.; Lazakis, I. Real-time data-driven missing data imputation for short-term sensor data of marine systems. A comparative study. *Ocean Eng.* **2020**, *218*, 108261. [[CrossRef](#)]
123. Merentitis, A.; Debes, C. Many hands make light work—On ensemble learning techniques for data fusion in remote sensing. *IEEE Geosci. Remote Sens. Mag.* **2015**, *3*, 86–99. [[CrossRef](#)]
124. Alshboul, O.; Almasabha, G.; Shehadeh, A.; Al-Shboul, K. A comparative study of LightGBM, XGBoost, and GEP models in shear strength management of SFRC-SBWS. *Structures* **2024**, *61*, 106009. [[CrossRef](#)]
125. Choubin, B.; Darabi, H.; Rahmati, O.; Sajedi-Hosseini, F.; Kløve, B. River suspended sediment modelling using the CART model: A comparative study of machine learning techniques. *Sci. Total. Environ.* **2018**, *615*, 272–281. [[CrossRef](#)] [[PubMed](#)]
126. Gillfeather-Clark, T.; Horrocks, T.; Holden, E.-J.; Wedge, D. A comparative study of neural network methods for first break detection using seismic refraction data over a detrital iron ore deposit. *Ore Geol. Rev.* **2021**, *137*, 104201. [[CrossRef](#)]
127. Jacob, M.; Reddy, G.S.H.; Rappai, C.; Kapoor, P.; Kolhekar, M. A Comparative Study of Supervised and Reinforcement Learning Techniques for the Application of Credit Defaulters. In Proceedings of the 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangaluru, India, 7 October 2022; pp. 1–6.
128. Mavrogiorgou, A.; Kiourtis, A.; Kleftakis, S.; Mavrogiorgos, K.; Zafeiropoulos, N.; Kyriazis, D. A Catalogue of Machine Learning Algorithms for Healthcare Risk Predictions. *Sensors* **2022**, *22*, 8615. [[CrossRef](#)]
129. Padhee, S.; Swygert, K.; Micir, I. Exploring Language Patterns in a Medical Licensure Exam Item Bank. In Proceedings of the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9 December 2021; pp. 503–508.
130. Abdulaal, A.; Patel, A.; Charani, E.; Denny, S.; Alqahtani, S.A.; Davies, G.W.; Mughal, N.; Moore, L.S. Comparison of deep learning with regression analysis in creating predictive models for SARS-CoV-2 outcomes. *BMC Med. Inform. Decis. Making* **2020**, *20*, 299. [[CrossRef](#)] [[PubMed](#)]
131. Zhao, L.; Wu, J. Performance comparison of supervised classifiers for detecting leukemia cells in high-dimensional mass cytometry data. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20 October 2017; pp. 3142–3146.
132. Moreno-Ibarra, M.-A.; Villuendas-Rey, Y.; Lytras, M.D.; Yáñez-Márquez, C.; Salgado-Ramírez, J.-C. Classification of diseases using machine learning algorithms: A comparative study. *Mathematics* **2021**, *9*, 1817. [[CrossRef](#)]
133. Venkata Durga Kiran, V.; Vinay Kumar, S.; Mudunuri, S.B.; Nookala, G.K.M. Comparative Study of Machine Learning Models to Classify Gene Variants of ClinVar. In *Data Management, Analytics and Innovation, Proceedings of ICDMAI 2020*; Springer: Singapore, 2020; Volume 2, pp. 2435–2443.
134. Mishra, N.; Patil, V.N. Machine Learning based Improved Automatic Diagnosis of Soft Tissue Tumors (STS). In Proceedings of the 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 25–27 November 2022; pp. 1–5.



135. Reiter, W. Co-occurrence balanced time series classification for the semi-supervised recognition of surgical smoke. *Int. J. Comput. Assist. Radiol. Surg.* **2021**, *16*, 2021–2027. [[CrossRef](#)] [[PubMed](#)]
136. Baker, M.R.; Utku, A. Unraveling user perceptions and biases: A comparative study of ML and DL models for exploring twitter sentiments towards ChatGPT. *J. Eng. Res.* **2023**, *in press*. [[CrossRef](#)]
137. Fergani, B. Evaluating C-SVM, CRF and LDA classification for daily activity recognition. In Proceedings of the 2012 International Conference on Multimedia Computing and Systems, Tangiers, Morocco, 10 May 2012; pp. 272–277.
138. Zhang, B.H.; Lemoine, B.; Mitchell, M. Mitigating unwanted biases with adversarial learning. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA, 2 February 2018.
139. Hong, J.; Zhu, Z.; Yu, S.; Wang, Z.; Dodge, H.H.; Zhou, J. Federated adversarial debiasing for fair and transferable representations. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14 August 2021; pp. 617–627.
140. Zafar, M.B.; Valera, I.; Gomez-Rodriguez, M.; Gummadi, K.P. Fairness constraints: A flexible approach for fair classification. *J. Mach. Learn. Res.* **2019**, *20*, 1–42.
141. Zafar, M.B.; Valera, I.; Rognier, M.G.; Gummadi, K.P. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*; PMLR: Fort Lauderdale, FL, USA, 2017; pp. 962–970.
142. Feldman, M.; Friedler, S.A.; Moeller, J.; Scheidegger, C.; Venkatasubramanian, S. Certifying and removing disparate impact. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10 August 2015; pp. 259–268.
143. Goh, G.; Cotter, A.; Gupta, M.; Friedlander, M.P. Satisfying real-world goals with dataset constraints. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2415–2423.
144. Barocas, S.; Selbst, A.D. Big data’s disparate impact. *Calif. L. Rev.* **2016**, *104*, 671. [[CrossRef](#)]
145. Creager, E.; Madras, D.; Jacobsen, J.H.; Weis, M.; Swersky, K.; Pitassi, T.; Zemel, R. Flexibly fair representation learning by disentanglement. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9 June 2019; pp. 1436–1445.
146. Gupta, U.; Ferber, A.M.; Dilkina, B.; Steeg, G.V. Controllable guarantees for fair outcomes via contrastive information estimation. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 7610–7619. [[CrossRef](#)]
147. Quadrianto, N.; Sharmanska, V.; Thomas, O. Discovering fair representations in the data domain. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15 June 2019; pp. 8227–8236.
148. Mehrabi, N.; Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; Galstyan, A. A survey on bias and fairness in machine learning. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [[CrossRef](#)]
149. Zhou, B.-C.; Han, C.-Y.; Guo, T.-D. Convergence of stochastic gradient descent in deep neural network. *Acta Math. Appl. Sin. Engl. Ser.* **2021**, *37*, 126–136. [[CrossRef](#)]
150. Joo, G.; Park, C.; Im, H. Performance evaluation of machine learning optimizers. *J. IKEEE* **2020**, *24*, 766–776.
151. Si, T.N.; Van Hung, T. Hybrid Recommender System Combined Sentiment Analysis with Incremental Algorithm. In Proceedings of the 2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD), Danang, Vietnam, 4 August 2022; pp. 104–108.
152. Qian, J.; Wu, Y.; Zhuang, B.; Wang, S.; Xiao, J. Understanding gradient clipping in incremental gradient methods. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 13 April 2021; pp. 1504–1512.
153. Mai, V.V.; Johansson, M. Stability and convergence of stochastic gradient clipping: Beyond lipschitz continuity and smoothness. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18 July 2021; pp. 7325–7335.
154. Polyak, B. Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* **1964**, *4*, 1–17. [[CrossRef](#)]
155. Wilson, A.C.; Recht, B.; Jordan, M.I. A Lyapunov analysis of momentum methods in optimization. *arXiv* **2016**, arXiv:1611.02635.
156. Liu, C.; Belkin, M. Accelerating sgd with momentum for over-parameterized learning. *arXiv* **2018**, arXiv:1810.13395.
157. Nesterov, Y.E. A method of solving a convex programming problem with convergence rate  $O(k^{-2})$ . In *Doklady Akademii Nauk*; Russian Academy of Sciences: Moscow, Russia, 1983; Volume 269, pp. 543–547.
158. Gao, S.; Pei, Z.; Zhang, Y.; Li, T. Bearing fault diagnosis based on adaptive convolutional neural network with nesterov momentum. *IEEE Sens. J.* **2021**, *21*, 9268–9276. [[CrossRef](#)]
159. Xie, X.; Xie, X.; Zhou, P.; Li, H.; Lin, Z.; Yan, S. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv* **2022**, arXiv:2208.06677. [[CrossRef](#)]
160. GitHub—Adan. Available online: <https://github.com/sail-sg/Adan> (accessed on 22 March 2024).
161. Guan, L. AdaPlus: Integrating Nesterov Momentum and Precise Stepsize Adjustment on Adamw Basis. In Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14 April 2024; pp. 5210–5214.
162. GitHub—AdaPlus. Available online: <https://github.com/guanleics/AdaPlus> (accessed on 22 March 2024).
163. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
164. Zhang, N.; Lei, D.; Zhao, J.F. An improved Adagrad gradient descent optimization algorithm. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi’an, China, 25 June 2018; pp. 2359–2362.

165. Gaiceanu, T.; Pastravanu, O. On CNN Applied to Speech-to-Text—Comparative Analysis of Different Gradient Based Optimizers. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Virtual, 19 May 2021; pp. 000085–000090.
166. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
167. Sethi, B.; Goel, R. Exploring Adaptive Learning Methods for Convex Optimization. 2015. Available online: [https://www.deepmusings.net/assets/AML\\_Project\\_Report.pdf](https://www.deepmusings.net/assets/AML_Project_Report.pdf) (accessed on 20 February 2024).
168. Guo, J.; Baharvand, A.; Tazeddinova, D.; Habibi, M.; Safarpour, H.; Roco-Videla, A.; Selmi, A. An intelligent computer method for vibration responses of the spinning multi-layer symmetric nanosystem using multi-physics modeling. *Eng. Comput.* **2022**, *38* (Suppl. S5), 4217–4238. [[CrossRef](#)]
169. Agarwal, A.K.; Kiran, V.; Jindal, R.K.; Chaudhary, D.; Tiwari, R.G. Optimized Transfer Learning for Dog Breed Classification. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 18–22.
170. Hinton, G.; Srivastava, N.; Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on* **2012**, *14*, 2.
171. Huk, M. Stochastic optimization of contextual neural networks with RMSprop. In *Intelligent Information and Database Systems: 12th Asian Conference, ACIIDS 2020, Phuket, Thailand, March 23–26, 2020, Proceedings, Part II 12*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 343–352.
172. Elshamy, R.; Abu-Elnasr, O.; Elhoseny, M.; Elmougy, S. Improving the efficiency of RMSProp optimizer by utilizing Nesterov in deep learning. *Sci. Rep.* **2023**, *13*, 8814. [[CrossRef](#)] [[PubMed](#)]
173. Funk, S. RMSprop Loses to SMORMS3-Beware the Epsilon! Available online: <http://sifter.org/simon/journal/20150420> (accessed on 22 March 2024).
174. Rossbroich, J.; Gygax, J.; Zenke, F. Fluctuation-driven initialization for spiking neural network training. *Neuromorphic Comput. Eng.* **2022**, *2*, 044016. [[CrossRef](#)]
175. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
176. Llugsi, R.; El Yacoubi, S.; Fontaine, A.; Lupera, P. Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Netw. for the Andean city of Quito. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 12 October 2021; pp. 1–6.
177. Bellido-Jiménez, J.A.; Estévez, J.; García-Marín, A.P. New machine learning approaches to improve reference evapotranspiration estimates using intra-daily temperature-based variables in a semi-arid region of Spain. *Agric. Water Manag.* **2021**, *245*, 106558. [[CrossRef](#)]
178. Rozante, J.R.; Ramirez, E.; Ramirez, D.; Rozante, G. Improved frost forecast using machine learning methods. *Artif. Intell. Geosci.* **2023**, *4*, 164–181. [[CrossRef](#)]
179. Shafie, M.R.; Khosravi, H.; Farhadpour, S.; Das, S.; Ahmed, I. A cluster-based human resources analytics for predicting employee turnover using optimized Artificial Neural Network and data augmentation. *Decis. Anal. J.* **2024**, *11*, 100461. [[CrossRef](#)]
180. Ampofo, K.A.; Owusu, E.; Appati, J.K. Performance Evaluation of LSTM Optimizers for Long-Term Electricity Consumption Prediction. In Proceedings of the 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 19 September 2022; pp. 1–6.
181. Aguilar, D.; Riofrio, D.; Benitez, D.; Perez, N.; Moyano, R.F. Text-based CAPTCHA vulnerability assessment using a deep learning-based solver. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 12 October 2021; pp. 1–6.
182. Indolia, S.; Nigam, S.; Singh, R. An optimized convolution neural network framework for facial expression recognition. In Proceedings of the 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 26 November 2021; Volume 6, pp. 93–98.
183. Shuvo, M.M.H.; Hassan, O.; Parvin, D.; Chen, M.; Islam, S.K. An optimized hardware implementation of deep learning inference for diabetes prediction. In Proceedings of the 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Virtual, 17 May 2021; pp. 1–6.
184. Poorani, S.; Kalaiselvi, S.; Aarthi, N.; Agalya, S.; Malathy, N.R.; Abitha, M. Epileptic seizure detection based on hyperparameter optimization using EEG data. In Proceedings of the 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 23 March 2023; pp. 890–893.
185. Acharya, T.; Annamalai, A.; Chouikha, M.F. Efficacy of CNN-bidirectional LSTM hybrid model for network-based anomaly detection. In Proceedings of the 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 20 May 2023; pp. 348–353.
186. Mavrogiorgos, K.; Kiourtis, A.; Mavrogiorgou, A.; Gucek, A.; Menychtas, A.; Kyriazis, D. Mitigating Bias in Time Series Forecasting for Efficient Wastewater Management. In Proceedings of the 2024 7th International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 17 July 2024; pp. 185–190.
187. Ying, X. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1168, p. 022022.
188. Brinkmann, E.M.; Burger, M.; Rasch, J.; Sutour, C. Bias reduction in variational regularization. *J. Math. Imaging Vis.* **2017**, *59*, 534–566. [[CrossRef](#)]

189. Domingos, P. A unified bias-variance decomposition. In Proceedings of the 17th International Conference on Machine Learning; Morgan Kaufmann Stanford, Stanford, CA, USA, 29 June 2020; pp. 231–238.
190. Geman, S.; Bienenstock, E.; Doursat, R. Neural Netw. and the Bias/Variance Dilemma. *Neural Comput.* **1992**, *4*, 1–58. [CrossRef]
191. Neal, B.; Mittal, S.; Baratin, A.; Tantia, V.; Scicluna, M.; Lacoste-Julien, S.; Mitliagkas, I. A modern take on the bias-variance tradeoff in neural networks. *arXiv*, 2018; arXiv:1810.08591.
192. Osborne, M.R.; Presnell, B.; Turlach, B.A. On the lasso and its dual. *J. Comput. Graph. Stat.* **2000**, *9*, 319–337. [CrossRef]
193. Melkumova, L.E.; Shatskikh, S.Y. Comparing Ridge and LASSO estimators for data analysis. *Procedia Eng.* **2017**, *201*, 746–755. [CrossRef]
194. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2005**, *67*, 301–320. [CrossRef]
195. Van Dyk, D.A.; Meng, X.L. The art of data augmentation. *J. Comput. Graph. Stat.* **2001**, *10*, 1–50. [CrossRef]
196. Shorten, C.; Khoshgoftaar, T.M.; Furht, B. Text data augmentation for deep learning. *J. Big Data* **2021**, *8*, 101. [CrossRef] [PubMed]
197. Feng, S.Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; Hovy, E. A survey of data augmentation approaches for NLP. *arXiv* **2021**, arXiv:2105.03075.
198. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
199. Jaipuria, N. Deflating dataset bias using synthetic data augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14 June 2020; pp. 772–773.
200. Kim, E.; Lee, J.; Choo, J. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10 October 2021; pp. 14992–15001.
201. Iosifidis, V.; Ntoutsis, E. Dealing with bias via data augmentation in supervised learning scenarios. *Jo Bates Paul D. Clough Robert Jäschke* **2018**, *24*. Available online: <https://www.kbs.uni-hannover.de/~ntoutsis/papers/18.BIAS.pdf> (accessed on 20 February 2024).
202. McLaughlin, N.; Del Rincon, J.M.; Miller, P. Data-augmentation for reducing dataset bias in person re-identification. In Proceedings of the 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Karlsruhe, Germany, 25 August 2015; pp. 1–6.
203. Prechelt, L. Early stopping-but when? In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 55–69.
204. Li, M.; Soltanolkotabi, M.; Oymak, S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 26 August 2020; pp. 4313–4324.
205. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [CrossRef]
206. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
207. Hanson, S.; Pratt, L. Comparing biases for minimal network construction with back-propagation. *Adv. Neural Inf. Process. Syst.* **1988**, *1*, 177–185.
208. Tessier, H.; Gripon, V.; Léonardon, M.; Arzel, M.; Hannagan, T.; Bertrand, D. Rethinking Weight Decay for Efficient Neural Network Pruning. *J. Imaging* **2022**, *8*, 64. [CrossRef] [PubMed]
209. Nakamura, K.; Hong, B.W. Adaptive weight decay for deep neural networks. *IEEE Access* **2019**, *7*, 118857–118865. [CrossRef]
210. Deshpande, S.; Shuttleworth, J.; Yang, J.; Taramonli, S.; England, M. PLIT: An alignment-free computational tool for identification of long non-coding RNAs in plant transcriptomic datasets. *Comput. Biol. Med.* **2019**, *105*, 169–181. [CrossRef] [PubMed]
211. Hekayati, J.; Rahimpour, M.R. Estimation of the saturation pressure of pure ionic liquids using MLP artificial neural networks and the revised isofugacity criterion. *J. Mol. Liq.* **2017**, *230*, 85–95. [CrossRef]
212. Poernomo, A.; Kang, D.-K. Biased dropout and crossmap dropout: Learning towards effective dropout regularization in convolutional neural network. *Neural Netw.* **2018**, *104*, 60–67. [CrossRef] [PubMed]
213. Krishnaveni, K. A novel framework using binary attention mechanism based deep convolution neural network for face emotion recognition. *Meas. Sens.* **2023**, *30*, 100881. [CrossRef]
214. Li, X.; Grandvalet, Y.; Davoine, F.; Cheng, J.; Cui, Y.; Zhang, H.; Belongie, S.; Tsai, Y.-H.; Yang, M.-H. Transfer learning in computer vision tasks: Remember where you come from. *Image Vis. Comput.* **2020**, *93*, 103853. [CrossRef]
215. Koeshidayatullah, A. Optimizing image-based deep learning for energy geoscience via an effortless end-to-end approach. *J. Pet. Sci. Eng.* **2022**, *215*, 110681. [CrossRef]
216. Scardapane, S.; Comminiello, D.; Hussain, A.; Uncini, A. Group sparse regularization for deep neural networks. *Neurocomputing* **2017**, *241*, 81–89. [CrossRef]
217. Deakin, M.; Bloomfield, H.; Greenwood, D.; Sheehy, S.; Walker, S.; Taylor, P.C. Impacts of heat decarbonization on system adequacy considering increased meteorological sensitivity. *Appl. Energy* **2021**, *298*, 117261. [CrossRef]
218. Belaid, F.; Roubaud, D.; Galariotis, E. Features of residential energy consumption: Evidence from France using an innovative multilevel modelling approach. *Energy Policy* **2019**, *125*, 277–285. [CrossRef]
219. Kong A Siou, L.; Johannet, A.; Valérie, B.E.; Pistre, S. Optimization of the generalization capability for rainfall-runoff modeling by neural networks: The case of the Lez aquifer (southern France). *Environ. Earth Sci.* **2012**, *65*, 2365–2375. [CrossRef]



220. Shimomura, Y.; Komukai, S.; Kitamura, T.; Sobue, T.; Yamasaki, S.; Kondo, T.; Mizuno, S.; Harada, K.; Doki, N.; Tanaka, M.; et al. Identifying the Optimal Conditioning Intensity of Hematopoietic Stem Cell Transplantation in Patients with Acute Myeloid Leukemia in Complete Remission. *Blood* **2023**, *142*, 2150. [CrossRef]
221. Yoon, K.; You, H.; Wu, W.Y.; Lim, C.Y.; Choi, J.; Boss, C.; Ramadan, A.; Popovich, J.M., Jr.; Cholewicki, J.; Reeves, N.P.; et al. Regularized nonlinear regression for simultaneously selecting and estimating key model parameters: Application to head-neck position tracking. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104974. [CrossRef]
222. Lawrence, A.J.; Stahl, D.; Duan, S.; Fennema, D.; Jaeckle, T.; Young, A.H.; Dazzan, P.; Moll, J.; Zahn, R. Neurocognitive measures of self-blame and risk prediction models of recurrence in major depressive disorder. *Biol. Psychiatry Cogn. Neurosci. Neuroimaging* **2022**, *7*, 256–264. [CrossRef]
223. Kauttonen, J.; Hlushchuk, Y.; Tikka, P. Optimizing methods for linking cinematic features to fMRI data. *NeuroImage* **2015**, *110*, 136–148. [CrossRef] [PubMed]
224. Algamal, Z.Y.; Lee, M.H. Regularized logistic regression with adjusted adaptive elastic net for gene selection in high dimensional cancer classification. *Comput. Biol. Med.* **2015**, *67*, 136–145. [CrossRef] [PubMed]
225. Hussain, S.; Anwar, S.M.; Majid, M. Segmentation of glioma tumors in brain using deep convolutional neural network. *Neurocomputing* **2018**, *282*, 248–261. [CrossRef]
226. Peng, H.; Gong, W.; Beckmann, C.F.; Vedaldi, A.; Smith, S.M. Accurate brain age prediction with lightweight deep neural networks. *Med. Image Anal.* **2021**, *68*, 101871. [CrossRef]
227. Vidya, B.; Sasikumar, P. Parkinson’s disease diagnosis and stage prediction based on gait signal analysis using EMD and CNN-LSTM network. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105099. [CrossRef]
228. Zhong, R.; Xie, X.; Luo, J.; Pan, T.; Lam, W.; Sumalee, A. Modeling double time-scale travel time processes with application to assessing the resilience of transportation systems. *Transp. Res. Part B Methodol.* **2020**, *132*, 228–248. [CrossRef]
229. Jenelius, E. Personalized predictive public transport crowding information with automated data sources. *Transp. Res. Part C Emerg. Technol.* **2020**, *117*, 102647. [CrossRef]
230. Tang, L.; Zhou, L.; Song, P.X.-K. Distributed simultaneous inference in generalized linear models via confidence distribution. *J. Multivar. Anal.* **2019**, *176*, 104567. [CrossRef] [PubMed]
231. Ma, W.; Qian, Z. Statistical inference of probabilistic origin-destination demand using day-to-day traffic data. *Transp. Res. Part C: Emerg. Technol.* **2018**, *88*, 227–256. [CrossRef]
232. Wu, J.; Zou, D.; Braverman, V.; Gu, Q. Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate. *arXiv* **2020**, arXiv:2011.02538.
233. Yildirim, H.; Özkale, M.R. The performance of ELM based ridge regression via the regularization parameters. *Expert Syst. Appl.* **2019**, *134*, 225–233. [CrossRef]
234. Abdulhafedh, A. Comparison between common statistical modeling techniques used in research, including: Discriminant analysis vs. logistic regression, ridge regression vs. LASSO, and decision tree vs. random forest. *OALib* **2022**, *9*, 1–19. [CrossRef]
235. Slatton, T.G. *A Comparison of Dropout and Weight Decay for Regularizing Deep Neural Networks*; University of Arkansas: Fayetteville, AR, USA, 2014.
236. Holroyd, J.; Scaife, R.; Stafford, T. What is implicit bias? *Philos. Compass* **2017**, *12*, e12437. [CrossRef]
237. Oswald, M.E.; Grosjean, S. Confirmation bias. Cognitive illusions: A handbook on fallacies and biases in thinking. *Judgem. Memory* **2004**, *79*, 83.
238. Winter, L.C. Mitigation and Prediction of the Confirmation Bias in Intelligence Analysis. 2017. Available online: [https://www.researchgate.net/profile/Lisa-Christina-Winter/publication/321309639\\_Mitigation\\_and\\_Prediction\\_of\\_the\\_Confirmation\\_Bias\\_in\\_Intelligence\\_Analysis/links/5b92513aa6fdccfd541fe3e0/Mitigation-and-Prediction-of-the-Confirmation-Bias-in-Intelligence](https://www.researchgate.net/profile/Lisa-Christina-Winter/publication/321309639_Mitigation_and_Prediction_of_the_Confirmation_Bias_in_Intelligence_Analysis/links/5b92513aa6fdccfd541fe3e0/Mitigation-and-Prediction-of-the-Confirmation-Bias-in-Intelligence) (accessed on 20 February 2024).
239. Heuer, R.J. Psychology of Intelligence Analysis; Center for the Study of Intelligence. 1999. Available online: [https://books.google.gr/books?hl=en&lr=&id=rRXFhKAIg8gC&oi=fnd&pg=PR7&dq=Psychology+of+Intelligence+Analysis&ots=REPkPSAYsO&sig=EghU1UDFes1BiaFHTpdYyOvWNng&redir\\_esc=y#v=onepage&q=Psychology%20of%20Intelligence%20Analysis&f=false](https://books.google.gr/books?hl=en&lr=&id=rRXFhKAIg8gC&oi=fnd&pg=PR7&dq=Psychology+of+Intelligence+Analysis&ots=REPkPSAYsO&sig=EghU1UDFes1BiaFHTpdYyOvWNng&redir_esc=y#v=onepage&q=Psychology%20of%20Intelligence%20Analysis&f=false) (accessed on 20 February 2024).
240. Lord, C.G.; Lepper, M.R.; Preston, E. Considering the opposite: A corrective strategy for social judgment. *J. Personal. Soc. Psychol.* **1984**, *47*, 1231. [CrossRef] [PubMed]
241. Romano, S.; Fucci, D.; Scanniello, G.; Baldassarre, M.T.; Turhan, B.; Juristo, N. On researcher bias in Software Engineering experiments. *J. Syst. Softw.* **2021**, *182*, 111068. [CrossRef]
242. Biderman, S.; Scheirer, W.J. Pitfalls in Machine Learning Research: Reexamining the Development Cycle. 2020. Available online: <https://proceedings.mlr.press/v137/biderman20a> (accessed on 20 February 2024).
243. Pinto, N.; Doukhan, D.; DiCarlo, J.J.; Cox, D.D. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput. Biol.* **2009**, *5*, e1000579. [CrossRef] [PubMed]
244. Liu, X.; Faes, L.; Kale, A.U.; Wagner, S.K.; Fu, D.J.; Bruynseels, A.; Mahendiran, T.; Moraes, G.; Shamdas, M.; Kern, C.; et al. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: A systematic review and meta-analysis. *Lancet Digit. Health* **2019**, *1*, e271–e297. [CrossRef] [PubMed]
245. Bellamy, R.K.; Bellamy, R.K.; Dey, K.; Hind, M.; Hoffman, S.C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; et al. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J. Res. Dev.* **2019**, *63*, 4:1–4:15. [CrossRef]

246. Bird, S.; Dudík, M.; Edgar, R.; Horn, B.; Lutz, R.; Milan, V.; Sameki, M.; Wallach, H.; Walker, K. Fairlearn: A Toolkit for Assessing and Improving Fairness in AI. Microsoft, Tech. Rep. MSR-TR-2020-32. 2020. Available online: [https://www.microsoft.com/en-us/research/uploads/prod/2020/05/Fairlearn\\_WhitePaper-2020-09-22.pdf](https://www.microsoft.com/en-us/research/uploads/prod/2020/05/Fairlearn_WhitePaper-2020-09-22.pdf) (accessed on 15 May 2024).
247. Johnson, B.; Brun, Y. Fairkit-learn: A fairness evaluation and comparison toolkit. In Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, Pittsburgh, PA, USA, 21 October 2022; pp. 70–74.
248. Hufthammer, K.T.; Aasheim, T.H.; Ånneland, S.; Brynjulfsen, H.; Slavkovik, M. Bias mitigation with AIF360: A comparative study. In Proceedings of the NIKT: Norsk IKT-Konferanse for Forskning og Utdanning, Virtual, 24 November 2020; Norsk IKT-Konferanse for Forskning og Utdanning: Norway, 2020.
249. Weerts, H.; Dudík, M.; Edgar, R.; Jalali, A.; Lutz, R.; Madaio, M. Fairlearn: Assessing and improving fairness of ai systems. *arXiv* **2023**, arXiv:2303.16626.
250. Gu, J.; Oelke, D. Understanding bias in machine learning. *arXiv* **2019**, arXiv:1909.01866.
251. Sengupta, E.; Garg, D.; Choudhury, T.; Aggarwal, A. Techniques to eliminate human bias in machine learning. In Proceedings of the 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 23 November 2018; pp. 226–230.
252. Hort, M.; Chen, Z.; Zhang, J.M.; Harman, M.; Sarro, F. Bias mitigation for machine learning classifiers: A comprehensive survey. *Acm, J. Respon. Comput.* **2023**, *1*, 1–52. [[CrossRef](#)]
253. Pagano, T.F.; Loureiro, R.B.; Lisboa, F.V.; Peixoto, R.M.; Guimarães, G.A.; Cruz, G.O.; Araujo, M.M.; Santos, L.L.; Cruz, M.A.; Oliveira, E.L.; et al. Bias and unfairness in machine learning models: A systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big Data Cognit. Comput.* **2023**, *7*, 15.
254. Suri, J.S.; Bhagawati, M.; Paul, S.; Protogeron, A.; Sfrikakis, P.P.; Kitas, G.D.; Khanna, N.N.; Ruzsa, Z.; Sharma, A.M.; Saxena, S.; et al. Understanding the bias in machine learning systems for cardiovascular disease risk assessment: The first of its kind review. *Comput. Biol. Med.* **2022**, *142*, 105204.
255. Li, F.; Wu, P.; Ong, H.H.; Peterson, J.F.; Wei, W.Q.; Zhao, J. Evaluating and mitigating bias in machine learning models for cardiovascular disease prediction. *J. Biomed. Inform.* **2023**, *138*, 104294. [[CrossRef](#)] [[PubMed](#)]
256. Zhang, K.; Khosravi, B.; Vahdati, S.; Faghani, S.; Nugen, F.; Rassoulinejad-Mousavi, S.M.; Moassefi, M.; Jagtap, J.M.M.; Singh, Y.; Rouzrokh, P.; et al. Mitigating bias in radiology machine learning: 2. Model development. *Radiol. Artif. Intell.* **2022**, *4*, e220010. [[CrossRef](#)] [[PubMed](#)]
257. Mavrogiorgou, A.; Kleftakis, S.; Mavrogiorgos, K.; Zafeiropoulos, N.; Menychtas, A.; Kiourtis, A.; Maglogiannis, I.; Kyriazis, D. beHEALTHIER: A microservices platform for analyzing and exploiting healthcare data. In Proceedings of the 34th International Symposium on Computer-Based Medical Systems, Virtual, 7 June 2021; pp. 283–288.
258. Biran, O.; Feder, O.; Moatti, Y.; Kiourtis, A.; Kyriazis, D.; Manias, G.; Mavrogiorgou, A.; Sgouros, N.M.; Barata, M.T.; Oldani, I.; et al. PolicyCLOUD: A prototype of a cloud serverless ecosystem for policy analytics. *Data Policy* **2022**, *4*. [[CrossRef](#)]
259. Kiourtis, A.; Poulakis, Y.; Karamolegkos, P.; Karabetian, A.; Voulgaris, K.; Mavrogiorgou, A.; Kyriazis, D. Diastema: Data-driven stack for big data applications management and deployment. *Int. J. Big Data Manag.* **2023**, *3*, 1–27. [[CrossRef](#)]
260. Reščič, N.; Alberts, J.; Altenburg, T.M.; Chinapaw, M.J.; De Nigro, A.; Fenoglio, D.; Gjoreski, M.; Gradišek, A.; Jurak, G.; Kiourtis, A.; et al. SmartCHANGE: AI-based long-term health risk evaluation for driving behaviour change strategies in children and youth. In Proceedings of the International Conference on Applied Mathematics & Computer Science, Lefkada Island, Greece, 8 August 2023; pp. 81–89.
261. Mavrogiorgou, A.; Kiourtis, A.; Makridis, G.; Kotios, D.; Koukos, V.; Kyriazis, D.; Soldatos, J.; Fatouros, G.; Drakoulis, D.; Maló, P.; et al. FAME: Federated Decentralized Trusted Data Marketplace for Embedded Finance. In Proceedings of the International Conference on Smart Applications, Communications and Networking, Istanbul, Turkey, 25 July 2023; pp. 1–6.
262. Manias, G.; Apostolopoulos, D.; Athanassopoulos, S.; Borotis, S.; Chatzimallis, C.; Chatzipantelis, T.; Compagnucci, M.C.; Drakslar, T.Z.; Fournier, F.; Goralczyk, M.; et al. AI4Gov: Trusted AI for Transparent Public Governance Fostering Democratic Values. In Proceedings of the 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Pafos, Cyprus, 19 June 2023; pp. 548–555.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.