Chapter 1

Fairness, Explainability, Privacy, and Robustness for Trustworthy Algorithmic Decision Making

Subhabrata Majumdar

ABSTRACT

With the rapid increase in use and deployment of machine learning (ML) systems in the world, concomitant concerns on the ethical implications of their downstream effect have surfaced in recent years. Responding to this challenge, the field of trustworthy ML has grown rapidly, and resulted in a large body of methods and algorithms that embody desirable qualities such as fairness, transparency, privacy and robustness. In this chapter, we survey the current landscape of trustworthy ML methods, introduce fundamental concepts, and summarize research directions. To bridge the gap between theory and practice, we provide implementation details of each category of methods that are currently available publicly.

KEYWORDS

Trustworthy machine learning, Fairness, ML bias, Explainable AI, Differential Privacy, Adversarial robustness

1 INTRODUCTION

In the current digital world, statistics and machine learning (ML) techniques that use large quantities of data are being deployed by companies across a broad range of industries. Especially in the last decade or so, this has been enabled by the increasing sophistication of high-performance computing and the democratization of data and data-analytical tools. While this automation of decision making has resulted in significant increase of efficiency and revenues in business processes, some unintentional negative impacts of such people-facing implementations have recently come to light. For example, Amazon had to abandon a ML system aimed at streamlining its hiring process by shortlisting resumes since it was discriminating against female applicants due to gender imbalance in historical data [1]. A number of targeting options in Facebook's advertising platform were correlated with sensitive features like gender and race [2]. As a result certain categories of targeted ads disproportionately left out minority groups. Several other examples exist that have put forth questions on

lapses in transparency, privacy or security, reliability and robustness [3].

Incorporating such value-driven qualities into ML systems has been the objective of a growing field of research in recent past. This field is often referred to by terms such as Trustworthy ML or responsible ML [4,5]. While the initial push started as a spurt of action among computer science researchers, due to the practical nature of problems tackled a number of interdisciplinary dimensions emerged soon to make the developed solutions relevant to actual stakeholders. Keeping this in mind, in this chapter we survey the landscape of trustworthy machine learning research and fundamental concepts. We devote each section of the chapter to one major aspect of trustworthy ML—fairness, transparency, privacy, and robustness. To focus on the implementation aspects of each method, at the end of a section we provide pointers to open-source computational resources for the interested reader.

2 FAIRNESS IN MACHINE LEARNING

In the context of big data and ML, the concepts of fairness and bias are heavily related, and may carry a number of implications based on the specific application. While there are many different kinds of bias (e.g. estimation bias, confirmation bias, cognitive bias) with not necessarily negative connotations, we focus on demographic bias due to inherent issues in the data and/or ML model outcomes that perpetuate historical and systemic inequities. Broadly speaking, fairness can be construed as the equal treatment of similar individuals—within or irrespective of whether they belong to specific demographic groups. In spite of being intuitive, some theoretical assumptions are required for a strict mathematical formulation of the above, and multiple definitions of fairness exist based on requirements of the specific application [6]. However a common conceptual thread running through such definitions is the shared objective that the deviation (or statistical bias) of one or more *parity metrics* should be minimized across individuals or individual groups of interest.

In this section, we present an overview of formal notions of such parity metrics and fairness definitions (Section 2.1), bias mitigation methodology (Section 2.2), and tools and frameworks to implement such methods in practice (Section 2.3). We summarize the overarching concepts at a fairly high level, owing to the fact that ML bias and fairness has been a hotly researched area in the recent past. Interested readers can check fairness-specific literature surveys for more granular discussions and references [6–8].

2.1 Fairness metrics and definitions

Parity measurement metrics quantify the extent a fairness notion is adhered to for an attribute or prediction outcome under consideration. While a number of such metrics exist in the literature [7,9], for brevity we define below some of the most widely used metrics. For all definitions, we denote by Y, X, S, $\hat{Y}$ the random

variables denoting respectively the binary output feature, input feature(s), sensitive feature and predicted output from a ML model. Also denote the probability of an event $A \in \mathcal{A}$, a set of events, by $P(A)$.

Definition 1. The prediction $\hat{Y}$ satisfies *equalized odds* [10] with respect to sensitive attribute S and output Y if

$$P(\hat{Y} = 1|S = 0, Y = y) = P(\hat{Y} = 1|S = 1, Y = y); \quad y = 0, 1.$$

In other words, $\hat{Y}$ and S are independent conditional on Y.

Definition 2. The prediction $\hat{Y}$ satisfies *demographic parity* [11] with respect to sensitive attribute S if

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

A similar definition with Y in place of $\hat{Y}$ denotes the demographic parity of the actual output feature values in the data (rather than output values predicted from the model).

Definition 3. The prediction $\hat{Y}$ is said to achieve *counterfactual fairness* [12] if for any value of the input(s), say $X = x$, the probability that $\hat{Y} = y$ for any y is the same across the values of S. This ensures that all individuals are treated similarly irrespective of their demographic group membership.

It is possible that a fairness metric conforms to only certain definitions of fairness. For example, equalized odds and demographic parity ensure *group fairness*, i.e. similar treatment/outcomes getting assigned to groups of people defined by one of more sensitive demographic features (such as race, gender). On the other hand, counterfactual fairness implies *individual fairness*, which refers to similar treatment of similar individuals irrespective of their sensitive feature values [11]. A third notion of fairness also exists at the intersection of these two, called *subgroup fairness*, and can involve simultaneously optimizing for multiple metrics [13,14].

2.2 Bias mitigation in ML models

While fairness concerns tend to stem from systemic problems and data quality issues, rectifying such problems is often nontrivial, cost-prohibitive and even impossible in practice [15]. Apropos of a typical ML model building pipeline (See Figure 1), research on bias-aware ML methods can be divided into three stages: pre-processing, in-processing, and post-processing. While fair versions of many ML or statistical techniques such as principal component analysis (PCA) [16], clustering [17], community detection [18], and causal models [19] exist in the literature, a disproportionate amount of research on fairness methodology is on supervised ML models [7]—owing possibly to their widespread use in real applications. For this reason, we summarize below the three stages assuming a supervised ML model with continuous or discrete outputs; see Table 1 for information on exemplar methods applicable to each of these stages.

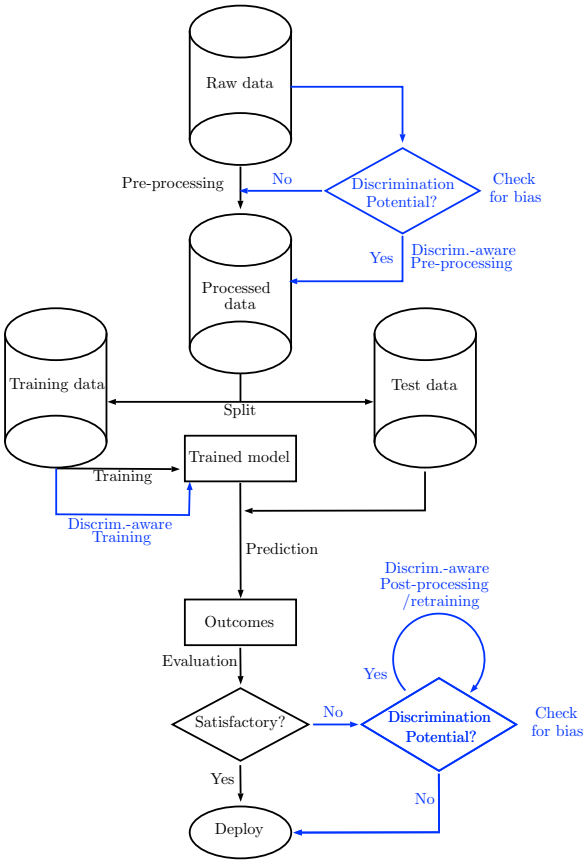

FIGURE 1 A general ML pipeline that shows opportunities for bias detection and mitigation in pre-, in-, and postprocessing stages (marked in blue).

Pre-processing

These model-agnostic methods aim to address fairness issues in data before it is fed into a ML model by transforming one or more features. Such transformations can either be pre-determined functions (DIR in Table 1), or learned from the data (LFR, OP), and can operate on the output Y and/or input X.

In-processing

Such algorithms incorporate one or more metrics directly into the model training process, utilizing techniques such as adversarial training [24] and regularization [26]. While conceptually these methods are fairly general, model-specific implementations may be very different. The metrics can be either fairness-specific (AD, PR) or related to general model performance (MA).

Stage	Technique	Problem Type	Data Required	Fairness Level	Applicable Fairness Metrics
Pre-processing	Disparate impact remover (DIR)[20]	Classification	S, X	Group	Disparate impact
	Learning fair representations (LFR)[21]	Classification	Y, S, X	Group Individual	Statistical parity
	Optimized pre-processing (OP)[22]	Classification	Y, S, X	Group Individual	General strategy
	Reweighing [23]	Classification	Y, S, X	Group	Statistical parity difference
In-processing	Adversarial debiasing (AD)[24]	Optimization	Y, S, X	Individual	Equality of odds, Demographic parity, Equality of opportunity
	Prejudice remover (PR)[25]	Optimization	Y, S, X	Individual	Prejudice index (PI), Normalized PI
	Meta-algorithm for fair classification (MA)[26]	Classification	Y, S, X	Group	Accuracy, precision, recall, True/false positive
Post-processing	Reject option classification (RO)[27]	Classification	$\hat{P}, S, X$	Group Individual	Equality of odds, Demographic parity, Equality of opportunity
	Equalized odds (EO)[10]	Classification, binary output	$\hat{Y}, S$	Group	Accuracy, precision, recall, True/false positive
	Calibrated equalized odds (CEO)[28]	Classification, probability output	$\hat{P}, S$	Group	AUC, lift, capture rate

TABLE 1 Mitigation algorithms for every stage of ML model building. For classification models with probability output, $\hat{P} \equiv P(\hat{Y} = 1)$.

Post-processing

Similar to pre-processing techniques, these methods do not require access to the trained model. However instead of the training data they operate on the predicted outputs of the ML model and attempt to mitigate bias in the predictions—ensured using equity in performance (EO, CEO) or fairness (RO) metrics. Such methods are particularly useful in third-party situations when the modeler does not have access to the training data, model or both.

2.3 Implementation

There are a number of toolkits, built on top of numerical programming languages such as Python and R, which package existing fairness-related methods aimed towards calculation of metrics and bias mitigation. AI Fairness 360 (AIF360)[11] is perhaps the most well-known of them. Among other such packages, Aequitas [29], Fairness Measures [30], FairML [31], FairTest [32], and Themis [33] are capable of bias detection, while Fairlearn [34] and Themis-ml [35] can perform both bias detection and mitigation. These packages are largely open-source—thus technically expandable to incorporate new metrics and mitigation techniques in an on-demand basis.

The above methods and packages provide parts of the technical apparatus to integrate fairness monitoring into different stages of the ML pipeline. However, implementing them into real-world projects is challenging. Besides the obvious

limitation of not being able to verify or fulfil theoretical conditions for individual detection and mitigation techniques in practice, several other challenges exist, with often domain-specific nuances [15,36]. In a survey of industry ML practitioners, [15] identified a number of such challenges as the main impediment to developing and deploying of fairness-aware ML processes:

1. Lack of guidance in data collection and identification sensitive features,
2. Blind spots in detecting bias concerns due to lack of team diversity and domain knowledge,
3. Use case diversity and lack of adequate tools for the specific project domain,
4. The need for human oversight for bias risk assessment in the different stages.

To address technical concerns in data collection, blind spots and lack of specific guidance, a number of recently proposed methods enable documentation and lineage tracking of ML lifecycles to aid in future reuse. These include Datasheets [37], data nutrition labels [38], FactSheets [39], and Model Cards [40]. To address scalability challenges of bias detection and mitigation in large scale ML workflows, the LinkedIn Fairness Toolkit [41] provides an open-source Scala/Spark library implementing a number of fairness metrics.

Effectively integrating human oversight for fairness-aware ML is a more challenging proposition. Depending on the fairness risk of a project and the potential adverse impact or such risks, this can necessarily be a deliberative and slow process. A number of human-in-the-loop strategies help in such situations. For example, structured algorithmic audits [42] and co-designed fairness checklists [43] can ensure that deployed ML models conform to company values and principles *and* meet performance metrics. Performing such risk assessments at multiple stages of the ML workflow, guided by documented information from past similar projects and the oversight of in-house subject matter experts [44] makes the final deployed system increasingly more likely to function in a responsible manner while satisfying business goals.

3 EXPLAINABLE ARTIFICIAL INTELLIGENCE

In high-stakes automated decision making, such as disease diagnosis or recidivism prediction, the need for explaining and elucidating decisions of the ML model involved is a crucial factor in eliciting the trust of stakeholders and regulatory authorities. However, owing to their complexity and scale, production-grade ML systems—mostly 'black-box' models that are easily amenable to experimentation but not explanation—suffer from lack of transparency to their inner workings that produce user-facing decisions [45–47]. Motivated by such needs, the field of Explaniable Artificial Intelligence (XAI) attempts to deal with the broad problem of comprehending an ML model and its (potential) predictions.

XAI consists of the study and research on several related but distinct concepts, such as interpretability, explainability, intelligibility, all pertaining to making ML models more comprehensible to stakeholders and end-users. To

clarify this ambiguity, we begin with an overview of formalisms of the concepts (Section 3.1). Following this we review the major technical concepts and their applications (Section 3.2), and finish with an impact assessment of XAI methods (Section 3.3). Similar to Section 2, we keep the discourse at a high level, and refer the interested reader to a number of high-quality resources for literature surveys [46–50] and conceptual details [51,52].

3.1 Formal objectives of XAI

Following philosophical notions of what constitutes an explanation [53], and their interpretations in the context of ML, explanability of a ML model refers to the answers to 'why' questions based on its existing or potential predicted outcomes [46,48]. Such answers need to achieve both *interpretability* and *completeness*. In other words, an answer needs to be (a) *comprehensible*, i.e. good enough to explain the mechanisms of a potentially complex ML model to a potentially non-technical audience, and (b) *correct*, i.e. an accurate enough description of how the model actually works. This is not an easy task. While one simple sweeping answer to diverse 'why' questions is easily comprehensible, it may not be an accurate representation of a model and can even be overly *persuasive* to elicit undue trust of a human evaluator [54]. On the other hand, accurately explaining the complexities and edge cases of a predictive model runs the risk of information overload. At the heart of effective explanation methods is a tradeoff between these two objectives [46,48,50,52]—customized to the problem (or problem domain) at hand and the target audience of the explanation.

Why explain?

There are a number of (not necessarily disjoint) motivations to develop explainability techniques. The first three are due to the survey on XAI methods by [47], while the fourth one is motivated by the need to move from a deductive to inductive reasoning of explanations [49,55].

1. **Justification**: explanations can help justify dubious or negative decisions, defend algorithmic decision-making, or comply with rules and regulations—such as the 'right to explanation' under the European Union General Data Protection Regulation (GDPR)[1], or credit reporting reason codes[2].
2. **Control and improvement:** insights into how a ML model is making predictions helps pinpoint the reasons behind anomalous or erroneous behavior, and enables efficient troubleshooting in future iterations.
3. **Discovery:** explanations can help discover the limitations or errors in our decision-making and enrich human knowledge by articulating insights on predictions where the model performs better than human benchmarks.

1. https://gdpr-info.eu
2. https://www.reasoncode.org/reasoncode101

4. **Causation**: designing and conducting experiments based on explanations of model outcomes, then analyzing that observational data has the potential to form and validate hypotheses on cause-effect relatioships—thus moving forward from typical association-based data analyses ([49,56], Section 3.2.3).

Terminologies

We conclude the XAI formalisms with reconcilliation of a few terms. In the XAI literature, multiple words are used interchangeably, such as 'interpretable' and 'explainable'. While the specific words do have slightly different connotations, in XAI parlance they often denote similar notions of model comprehension, along with terms such as understandability, comprehensibility and intelligibility. One factor of the different terms is the target audience and context. For example, user-group specific Google search trends suggest that technical ML community of researchers and practitioners prefer using the word 'interpretable', while 'explainable' is more preferred in public discourse [47]. For ease of narrative we use 'explainability' throughout the next section, and return to this topic in Section 3.3.

3.2 Taxonomy of methods

While explainability methods can be divided according to their applicability in the three stages of a ML pipeline (Figure 1), pre-model explainability closely maps to data explainability and transformations—consisting of unsupervised methods such as PCA and K-means [46]. In this review, we focus on recent developments of in-model and post-model XAI methodology.

3.2.1 In-model vs. post-model explanations

The above two categories are closely tied to models that are being explained. In-model explainability pertains to the explanation method being tied to the model *by definition*. White box/glass box ML models that have an open architecture thus possess in-model or *intrinsic* explainability. On the other hand, black-box model structures that are difficult or impossible to represent explicitly—because of issues like scale and propriety—are more amenable to being explained by post-model or *post-hoc* explainability techniques.

There is another dimension to this dichotomy: model-specific vs. model-agnostic. By definition intrinsic methods are model-specific—explanations generated by an explainable model (e.g. linear regression, decision trees, LASSO) are specific to that model. On the other hand, post-hoc explanation techniques *tend to be* model-agnostic [46,47]. Post-hoc techniques (e.g. LIME [57], SHAP [58], MUSE [59]) can thus be applied to models that are explainable themselves. Independence of the base model and the resulting modularity is a clear advantage of post-hoc methods. However, the use of a second model introduces another scope of error in the explanation process apart from errors pertaining to the

main model. In feature-rich large datasets, multiple models often exist that have similar prediction performances—the so called 'Rashomon effect' [60,61]. In such situations, depending on driving factors such as the consequences of a false prediction and access to domain knowledge, the development of explainable models can be pursued that achieve similar performance as a black-box model *and* satisfy a specific definition of interpretability relevant to the problem [61].

3.2.2 Global and local explanations

Intrinsic and post-hoc explainable methods can produce explanations at either global or local level. Global explanation methods aim to produce an overall *comprehensible* overview of a ML model. Such overviews often [46] take the form of feature summaries [57,58], model internals (linear models, LASSO, decision trees), representative data-points (tabular LIME), and surrogate models to explain explanations [59]. Note that the emphasis here is on a human being able to comprehend the overview. Thus, global explanations need not necessarily be holistic. They can be *modular* instead, in the sense that a model interpretation can be decomposed into chunks (subsets of samples and/or features) that hold specific meanings to the aimed user [51,62].

Local explanations aim to explain one single sample or small groups of samples. The general idea here is that on small neighborhoods in the sample space, variations in the behavior of a trained model is low. As a result, simple, interpretable supervised models can be trained on tightly clustered data-points, taking model predictions as labels. Following LIME [57] and SHAP [58], a large number of local methods have been proposed for producing post-hoc explanations of black-box models. As summarized by a survey of post-hoc methods [50], a common principle local explanation methods are based on is measuring the effect of removing one or more features from a model on its predictions.

3.2.3 Causal explainability

Conventional ML models are based solely on observational data, and are thus only able to infer *associations* instead of true cause-effect relationships between features. Causal explainability methods based on evaluating counterfactual situations—proposing and evaluating alternate model outcomes under alternate situations such as different input features, training setups—provide tools that can address such shortcomings [55,63,64].

A major conceptual framework in causal inference is that of Structural Causal Models (SCM).

Definition 4. A *Structural Causal Model* [64] is definited by the 4-tuple (X, U, f, P_u), where

- X is a finite set of endogenous variables that are usually observable,
- U is a finite set of exogenous variables that are usually unobserved or noise,

Category	Purpose	References
Causal explainability of models	Explain effects of model component(s) on its predictions	[66–70]
Counterfactual explanations	Generate explanations for outcomes in alternate input or training scenarios	[71–76]
Causal fairness	Use of explainable causal models to ensure fairness	[12,77–79]
Verifying causal relationships in data	Verify causal assumptions between features, ensure interpretability using causal inference	[80,81]

TABLE 2 Categorization of causal explainability methods.

- $f = \{f_1, \ldots, f_n\}$ is a set of functions representing causal mechanisms

$$x_i = f_i(Pa(x_i), u_i),$$

  for $x_i \in X; Pa(x_i) \subseteq (X \backslash x_i) \cup U$,
- P_u is a probability distribution over U.

Methods in causal explainability (and causal inference in general) are mainly based on modelling complex ML models such as DNNs as SCM, incorporating causal reasoning and human intelligibility [55,65]. Moraffah et al [55] divide such methods into four categories; we summarize them in Table 2.

3.3 Do explanations serve their purpose?

Model explanations need to make sense to whoever the explanations are aimed at. Based on the expertise level of these stakeholders/end users (such as data scientist, project manager, business leader) and the use case being analyzed, different explanation methods may be prioritized. Motivated by this reasoning, [52] asked the question that whether all XAI methods are 'equally interpretable', and proposed three avenues of evaluating them—grounded on (i) application, (ii) human user, and (iii) the function of the model. The first two involve a human in the loop in evaluating either the implementation of the explanation by a domain expert (application-grounded), or the explanations directly (human user grounded). The third one evaluates explanation methods, that have already received some form of human vetting, through specific functional metrics.

In practice however, there is admittedly a disconnect between research on proposing new explainability methods vs. assessing existing methods on the above criteria; just 5% of XAI methods proposed until now deal with impact assessment of XAI [46,47]. This is an important shortcoming. The inherent subjective nature of the area (that should be clear to the reader by now) means that *explainability lies in the eyes of the stakeholder*—it is imperative to ensure that an XAI method is actually serving its purpose of being useful to the users it is producing explanations for.

3.3.1 From explanation to understanding

To develop effective explanation methods that translates to understanding of a ML model and its actions for the human in the loop, insights from other fields like psychology and philosophy can be borrowed [82,83]. This means making an effort to produce explanations that are human-like or at the least human-friendly. To mimic the sparse and prototype-based nature of human reasoning, human-like explanations [84,85] need to be (i) *contrastive*: explain why one event happened instead of another, (ii) *selective*: avoid information overload by focusing on a small number of causes, and (iii) *relatable*: appeal to the mental model of explainee and let them draw inference. Human-friendly methods, on the other hand, tend to focus on producing intelligible interpretations and visualizations of complex ML models or their outcomes. Apart from ML expertise, conceptualizing and implementing such techniques often and should involve concepts from the field of Human-Computer Interaction (HCI) [86]. Relevant works in the intersection of XAI and HCI that build explanation interfaces include [87], eXplainable AI for Designers (XAID) [88], Rivelo [89], and Gamut [90]. The Weight of Evidence framework and meta-algorithm of [91] is one of the first attempts to produce explanations that are themselves human-oriented. Moving beyond model intelligibility, a number recent tools provide error analysis interfaces to help users explore the deficiencies of an ML model in detail [92–94].

The final step in ensuring that the products of XAI research and efforts of translating that to understanding of a model serve their purpose is user evaluation. While research in this work is extremely sparse, some very recent studies reveal interesting insights on the impact of XAI methods. A study [95] on data science practitioners comprising of a survey (sample size $N = 197$) and contextual enquiry ($N = 11$) revealed that users tend to put too much trust on automated explanations, and are inclined to trust a model based on its positive explanation without doing a detailed check. As is expected in this context, [96] showed that it is in fact possible to *mislead* users with explanations. Rogue black-box explanations generated using their proposed mechanism that *did not* include any sensitive features were able to successfully mislead domain experts into trusting a black-box model that actually used those sensitive features to make predictions. In perhaps the largest user study of its kind, [97] rigorously evaluated aspects of model explainability using a randomized experiment on 3800 participants. As part of a number of surprising outcomes, model interpretations seemed to make an user unduly believe in the efficacy of a badly performing model and correctness of its mis-predictions. Their observations also indicate that highly detailed explanations impede users' ability to detect unusual input feature values.

3.3.2 Implementations and tools

We conclude this section with an overview of available tools for the interested reader to implement XAI methods. Implementations of a number of well-known

methods are available in standard computer languages, such as in R[3], Python[4] and Julia[5]. Going a step further, two open-source projects also offer expandable platforms for practitioners to implement their own XAI methods, in addition to a number of built-in options for existing methods—AI Explainability 360 [98] by IBM and InterpretML [99] by Microsoft. A recent review article summarizes and compares a number of R packages for XAI [100]. Given the subjectivity and importance of the role of explainability in ML systems, these resources are valuable in driving the adoption of XAI methods in practice.

4 NOTIONS OF ALGORITHMIC PRIVACY

As ML algorithms get developed and deployed in the real world with increasing frequency, the longstanding problem of how to preserve the privacy of individuals in any data-analytic exercise has become more and more important. While traditional approaches based on anonymization are somewhat effective in concealing direct identifiers such as name or address, reidentification of subjects using auxiliary data available elsewhere is a very real threat [101,102]—more so in the twenty-first century world where access to information is democratized.

At a high level, algorithmic privacy aims to provide meaningful answers to specific questions (or 'queries') about the population of interest (or the representative sample) *without* disclosing any individual's information. A popular way of achieving this is to pose such queries to a trusted curator with full access to the data, which computes and releases an answer that is 'safe enough' according to a predefined privacy guarantee. Given the diverse domains (such as medical or social sciences, advertising, communication) or modalities (graph, streaming data, manifolds) large real-world datasets can be associated with, providing such secure answers is not easy in general.

While there are other ways to ensure inferential privacy and maintaining accuracy of the answer at the same time (such as [103]), in this section we focus on the area of *Differential Privacy* (DP) that has seen notable developments in the last decade. We begin with an overview of basic definitions and concepts (Section 4.1), then review major streams of DP research (Section 4.2). We finish with an overview of privacy frameworks that extend or build up on the concept of DP, and some real-world examples (Section 4.3). For mathematical and methodological details, we refer the interested reader to a number of technical resources [101,104–107].

3. https://uc-r.github.io/iml-pkg
4. https://www.analyticsvidhya.com/blog/2020/03/6-python-libraries-interpret-machine-learning-models
5. https://github.com/interpretable-ml/IML.jl

4.1 Preliminaries of differential privacy

The fundamental idea of DP is randomization—the curator introduces enough *data-independent* noise in the query output such that the noisy output is 'similar' to the original output, but does not give away information on the inclusion of individual data-points in computing the output. A fixed *privacy budget* $\epsilon > 0$ quantifies this similarity, with values closer to 0 denoting higher degrees of similarity. Noise distributions are specific to the class of queries being answered, and there is an inherent information loss that needs to be traded off against an individual's privacy risk. We formalize these shortly (see also Figure 2).

Consider datasets $X \in \mathcal{X}^n$ comprising of n samples, each drawn from domain $\mathcal{X}$, a function $M_q : \mathcal{Q} \times \mathcal{X}^n \to \mathcal{R}$ that takes in a query $q \in \mathcal{Q}$ on the dataset and gives a *randomized* answer in the range $\mathcal{R}$. Also, suppose $Y \in \mathcal{X}^n$ is another dataset that differs from X in one element—call it an adjacent dataset of X, and denote by $X \sim Y$.

Definition 5 ([108])**.** The mechanism M_q is called ϵ-Differentially Private, or ϵ-DP in short, if for any pair of adjacent datasets X, Y the following holds for any measurable $R \subseteq \mathcal{R}$:

$$P(M_q(X) \in R) \leq e^{\epsilon} P(M_q(Y) \in R). \tag{1.1}$$

The mechanism M_q is called (ϵ, δ)-Differentially Private, or (ϵ, δ)-DP, if:

$$P(M_q(X) \in R) \leq e^{\epsilon} P(M_q(Y) \in R) + \delta. \tag{1.2}$$

Notice that (ϵ, δ)-DP is a weaker condition than ϵ-DP, and ϵ-DP is the same as $(\epsilon, 0)$-DP. Typically, ϵ, δ are small but non-negligible, so that the difference between the two probabilities in (1.1) and (1.2) is minimal (Figure 2). Consequently, whether a particular sample belongs to X or not, the answer to q remains almost the same. A malicious analyst would thus not be able to get any information about this sample by observing the dissimilarities between $M_q(X)$ and $M_q(Y)$.

There are two broad classes of queries q: numeric and non-numeric queries, pertaining to numeric (such as mean, median, quantiles), or non-numeric (such as maximum, minimum, top 10%) answers to q, respectively. Formally, a query $q : \mathcal{X}^n \to \mathcal{R}$ maps a dataset to an output, and a mechanism M_q introduces noise in the result-generating procedure of the query. For numerical queries, two well-known procedures introduce noise as a post-processing step.

Definition 6. For a numeric query $q : \mathcal{X}^n \to \mathbb{R}^d, d > 0$ giving real-valued results, *Global Sensitivity* is defined as

$$\Delta_p(q) \equiv \Delta(q, \|\cdot\|_p) = \max_{X \sim Y} \|q(X) - q(Y)\|_p, \tag{1.3}$$

for the ℓ_p-norm $\|\cdot\|_p$ and the maximum taken over all possible adjacent X, Y.

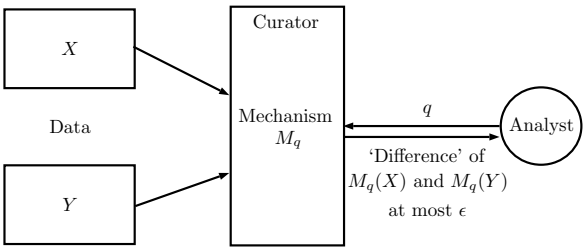

FIGURE 2 A schematic of Differential privacy. The answers to a query q, obtained using the randomized privacy algorithm $\mathcal{M}_q$, are very similar for two datasets X, Y which differ in only one sample.

Definition 7 ([108])**.** For a numeric query $q : \mathcal{X}^n \to \mathbb{R}^d$, the *Laplace Mechanism* is defined as $M_{q,\epsilon} \equiv M_q$ such that

$$M_q(X) = q(X) + (\eta_1, \ldots, \eta_d),$$

where $\eta_i, i = 1, \ldots, d$ are independently and identically distributed (hereafter i.i.d.) as Laplace$(\Delta_1(q)/\epsilon)$ random variables, with probability density function

$$p(\eta_i) \propto \exp\left(-\frac{\epsilon|\eta_i|}{\Delta_1(q)}\right).$$

Definition 8 ([109])**.** For a numeric query $q : \mathcal{X}^n \to \mathbb{R}^d$, the *Gaussian Mechanism* is defined as $M_{q,\epsilon,\delta} \equiv M_q$ such that

$$M_q(X) = q(X) + (\gamma_1, \ldots, \gamma_d),$$

where $\gamma_i, i = 1, \ldots, d$ are i.i.d. Gaussian random variables

$$N\left(0, \frac{2\log(2/\delta)\Delta_2^2(q)}{\epsilon^2}\right).$$

Given $\epsilon, \delta > 0$, the Laplace mechanism satisfies ϵ-DP, while the Gaussian mechanism satisfies (ϵ, δ)-DP.

The situation is somewhat complex for non-numeric queries due to their generic nature. Each (non-numeric) element $r \in \mathcal{R}$ is first assigned a utility score $u(X, r)$ specific to the dataset X, quantifying the preference of r as the answer to $q(X)$. While a direct (non-private) answer means choosing the r with the highest utility, privacy mechanisms randomize this choice. The *Exponential mechanism* [110] is one such well-known mechanism that preserves ϵ-DP.

Definition 9 ([110])**.** For a non-numeric query q with utility $u : \mathcal{X}^n \times \mathcal{R} \to \mathbb{R}$, the *Exponential mechanism* is defined as the randomized choice of an answer r:

$$P(M_q(X) = r) \propto \exp\left(\frac{\epsilon u(X, r)}{2\Delta(u)}\right),$$

with $\Delta(u) = \max_{r \in \mathcal{R}} \max_{X \sim Y} |u(X, r) - u(Y, r)|$ being the global sensitivity of the utility function.

Consequently, high utility answers are more likely to be chosen compared to those with lower utility. However the inherent randomness results in a privacy guarantee on the eventual answer.

Finally, combination of DP mechanisms is of practical interest, such as transformations on the mechanism M_q and answering multiple queries [101,104]. To this end, the following basic results hold:

Lemma 1 ([101])**.** If $M_q : \mathcal{Q} \times \mathcal{X}^n \to \mathcal{R}$ is (ϵ, δ)-DP, and $F : \mathcal{R} \to \mathcal{R}'$ is any random function, then $F \circ M_q : \mathcal{Q} \times \mathcal{X}^n \to \mathcal{R}'$ is (ϵ, δ)-DP.

Lemma 2 ([111])**.** If $M_{1q}, \ldots, M_{kq}, k > 0$ are (ϵ, δ)-DP mechanisms, then their serial combination $M_q = (M_{1q}, \ldots, M_{kq})$ is $(k\epsilon, k\delta)$-DP.

Lemma 3 ([111])**.** Given a dataset $X \in \mathcal{X}^n$, consider any partition $\{X_1, \ldots, X_k\}$, with $X_k \in \mathcal{X}^{n_k}$. If $M_{kq} : \mathcal{Q} \times \mathcal{X}^{n_k} \to \mathcal{R}$ are ϵ-DP mechanisms on the corresponding data partitions, then for any (fixed or random) function G, their parallel combination is ϵ-DP on the full dataset:

$$M_q(X) = G(M_{1q}(X_1), \ldots, M_{kq}(X_k)).$$

4.2 Privacy-preserving methodology

While the above principles provide privacy guarantees for broad classes of queries, implementing them for specific ML algorithms and model outputs is nontrivial. Laplace and Exponential mechanisms rely on the global sensitivity being a common upper bound to all possible numeric or non-numeric queries. For the specific dataset being analyzed, or the class of datasets that are of interest, this bound can be tightened—potentially reducing the injected random noise while still providing the same DP guarantees.

ML models often learn patterns from the data in an iterative process, with numerous calls to the training data. These calls can be seen as queries that may be combined or nested. For example, in any gradient descent-based optimization algorithm, each gradient computation can be seen as a query that can potentially be perturbed. These computations combine to produce a trained ML model, which can also be seen as a larger query that produces outputs or predictions. Depending on the specifics of the model or dataset, privacy noise injected at different stages of the training process can result in different magnitudes of perturbation in the final output.

4.2.1 Local sensitivity and other mechanisms

The concept of *Local sensitivity* [112] is based on the idea that tuning noise levels *locally* to the dataset being analyzed, instead of fixing a common upper bound (with global sensitivity) can potentially result in more accurate outputs with the same DP guarantee.

Definition 10. Consider a query $q \in \mathcal{Q} \rightarrow \mathcal{R}$, operating on a dataset X. Then we define the following.

- If q is numeric, the Local sensitivity (with ℓ_p norm) is

$$\Delta_p(q, X) = \max_{Y:X\sim Y} \|q(X) - q(Y)\|_p.$$

- If q is non-numeric, then for utility function u, the Local sensitivity is

$$\Delta(u, X) = \max_{r\in\mathcal{R}} \max_{Y:X\sim Y} |u(X, r) - u(Y, r)|.$$

Note that in the above definition the maximums are taken over all datasets Y adjacent to a fixed X, calibrating the noise magnitude to the data at hand. This is challenging, since the privacy noise level needs to be independent of the data, simply substituting the maximum value calculated from the samples in X does not satisfy DP guarantees [112,113]. Navigating this problem is comparatively easier for numeric queries—[112] proposed the smooth sensitivity framework and implemented it to design algorithms that solve k-means and Gaussian mixture learning problems while preserving (ϵ, δ)-DP. Building up on this fundamental work, local sensitivity-based methodology with DP guarantees have been proposed—such as in PCA [114], answering subgraph counting queries [115], and deep learning algorithms [116].

For non-numeric queries, the very recently proposed *Local dampening* mechanism incorporates local sensitivity to design private algorithms [113]. Their generic method uses attenuated versions of utility functions in combination with the Exponential mechanism, and is ϵ-DP. Compared to Exponential mechanism, a local dampening-based approach results in significant reduction of privacy budget in high-influence node detection problems on graphs, and higher accuracy in decision-tree based ML models [113].

A number of alternative mechanisms can be used in place of the traditional choices of Laplace/Gaussian/Exponential mechanisms. Bun and Steinke [117] proposed a local sensitivity framework that extends to three more noise distributions, as compared to only Laplace noise in smooth sensitivity [112]. Ladder functions [118] and the staircase mechanism [119] are alternatives to the Exponential mechanism for certain non-numeric queries. Very recently, [120] proposed a simple modification of the Exponential mechanism, called Permute-and-Flip, that significantly improves accuracy in private median estimation.

4.2.2 Algorithms with DP guarantees

References to DP answers for a number of simpler queries such as median, mean or distribution calculations, and broad classes of methods such as hypothesis testing and graph analysis can be found in [105]. For more complex ML models, there are two broad categories of perturbations, which add noise (i) directly into the training steps, or (ii) take the outputs or objective functions of a non-private model an perturb them [107].

DP versions of traditional 'shallow' ML and statistical models have been proposed and refined over the past few years [107]. This includes decision trees [109,121], Naïve bayes [122,123], bagging [124], random forest [125–127], clustering [112,128,129], PCA [109,114,130] and online learning [131,132]. At a more fundamental level, a number of papers focus on incorporating DP into generic computational algorithms that can be used in ML model training, for example Markov Chain Monte Carlo (MCMC) [133,134], Hamiltonian Monte Carlo [135], Expectation Maximization (EM) [136], and Stochastic Gradient Descent (SGD) [137–139]. All of these incorporate privacy perturbations into model training. Among methods that perturb model outputs/objective function, notable methods include DP generalized linear models [140] and their variants such as M-estimators [141] and LASSO [142], Support Vector Machines (SVM) [143,144], and Empirical Risk Minimization (ERM) in general [145–147].

Incorporating privacy guarantees into computation-intensive deep learning methods produce some unique challenges related to the large-scale nature of models, high noise during model training due to a large number of model parameters and the black-box nature of such models. For training-level perturbations, instead of using existing DP versions of SGD [137–139] it is possible to use distributed computing to speed up model training process *while* conforming to DP guarantees [148–150]. Moving beyond classification tasks, [151] proposed using Gaussian mechanism to ensure user-level privacy in Long Short-Term Memory (LSTM) Networks, while [152,153] use gradient perturbation to achieve DP in Generative Adversarial Networks (GAN). Finally, DP work on output perturbation of deep learning models is extremely scarce, owing to the instability of objective functions or final solution in this regime [107]. The three existing methods in this domain pertain to deep autoencoders [154], Convolutional deep belief networks [155], and deep network embeddings [156].

4.3 Generalizations, variants and applications

4.3.1 Pufferfish

The concept of Pufferfish [157] expands on the notions of DP to propose a larger class of privacy mechanisms that are able to counter many different types of malicious attacks. Based on domain knowledge, Pufferfish lets experts decide what secrets they want to protect, what secrets they wish to be indistinguishable, and what types of attacks they want to protect against. These three entities are formalized by the specifications of a Pufferfish framework:

- The set of secrets, denoted by $\mathcal{S}$,
- The set of secret pairs, denoted by $\mathcal{S}_P$,
- Data evolution scenarios: denoted by $\mathcal{D}$, a set of probability distributions over the data domain $\mathcal{X}^n$—representing the set of attackers to protect the secret pairs from.

Definition 11. Given a framework $(\mathcal{S}, \mathcal{S}_P, \mathcal{D})$ a mechanism M_q is ϵ-Pufferfish

for the query q if, for all distributions $\delta \in \mathcal{D}$ and a dataset X drawn from δ, all secret pairs $(s_i, s_j) \in \mathcal{S}_P$ such that $P(s_i|\delta) \neq 0, P(s_j|\delta) \neq 0$, and all $r \in \mathcal{R}$, we have

$$e^{-\epsilon} \leq \frac{P(M_q(X) = r)|s_i, \delta)}{P(M_q(X) = r)|s_j, \delta)} \leq e^{\epsilon}. \tag{1.4}$$

Note the obvious parallels with the definition of ϵ-DP (Definition 5). Indeed, DP is a special case of Pufferfish, where statements of the form 'sample i has value x', $i \in \{1, \ldots, n\}, x \in \mathcal{X}$ are the secrets, secret pairs are pairs of such statements for the same sample but different values, and $\mathcal{D}$ is composed of size-n data distributions with independent samples [157].

As the first practical instantiation of Pufferfish that is different from DP, [158] proposed the Wasserstein mechanism to ensure randomization-based privacy guarantees for correlated data (such as time series). Subsequent works on a similar theme include FGS-Pufferfish privacy for temporally correlated trajectories [159], private monitoring of web browsing activity [160] and Pufferfish for correlated categorical data [161]. In later research, the authors of Pufferfish proposed a further generalization called Blowfish [162] that allows to incorporate formal policies and data constraints to fine tune a privacy mechanism even more.

4.3.2 Other variations

Among other variations of the basic DP framework, perhaps the most intuitive and straightforward is that of Group Differential Privacy (GDP) [104]. Simply stated, instead of adjacent datasets, the definition of ϵ-GDP or (ϵ, δ)-GDP requires the same respective bounds in (1.1) or (1.2) hold over all datasets that differ in a number of prespecified group memberships of their samples. A number of applications of GDP propose privacy mechanisms for correlated data: such as network data [163], temporal correlations [164] and multilevel graphs [165].

Pufferfish formalizes the role of data distributions in the study of privacy. Alternatively, *divergences*—the concept of 'distances' between probability distributions—can be used to embed knowledge or assumptions on data distributions into privacy definitions. Rényi Differential Privacy [166] is one such method that uses the Rényi Divergence to propose a relaxation of conventional DP:

Definition 12 ([166])**.** For two probability distributions P, Q taking values in the range of query results $\mathcal{R}$, and $\alpha \geq 1$, define the *Rényi divergence* as

$$D_\alpha(P\|Q) = \begin{cases} \mathbb{E}_{x\sim P} \log \frac{P(x)}{Q(x)}, & \text{if } \alpha = 1, \\ \frac{1}{\alpha-1} \log \mathbb{E}_{x\sim Q} \left(\frac{P(x)}{Q(x)}\right)^\alpha & \text{if } \alpha > 1, \\ \sup_{x\in\mathcal{R}} \log \frac{P(x)}{Q(x)}, & \text{if } \alpha = \infty. \end{cases}$$

A mechanism M_q is said to satisfy ϵ-Rényi DP of order α, (α, ϵ)-Rényi DP in

Library	Platform	Link
Diffprivlib, by IBM	Python	https://github.com/IBM/differential-privacy-library
diffpriv	R	https://cran.r-project.org/web/packages/diffpriv/index.html
Google's differential privacy library	C++, Go, Java	https://github.com/google/differential-privacy
Opacus, by Facebook	Python (PyTorch)	https://github.com/pytorch/opacus
SmartNoise Core	Python	https://github.com/opendifferentialprivacy/smartnoise-core
WhiteNoise	Microsoft Azure ML	https://medium.com/microsoftazure/whitenoise-an-open-source-library-to-protect-data-with-differential-privacy-fed740e29a49
OpenDP	Python	https://github.com/opendifferentialprivacy

TABLE 3 Open-source tools on DP.

short, if for any pair of adjacent datasets X, Y the following holds:

$$D_\alpha(M_q(X)\|M_q(Y)) \leq \epsilon.$$

Compared to traditional DP, this relaxed definition provides better privacy guarantees under composition of multiple heterogeneous queries [166].

Other proposals that extend DP using notions related to data distributions include Concentrated DP [167,168], Capacity Bounded DP [169], Poisson Subsampled Rényi DP [170], Bootstrap DP [171] and f-DP [172]. Going in a different route than the divergence-based notion, the f-DP framework [172] focuses on a hypothesis testing interpretation of DP guarantees and implements privacy amplification methods using subsampling.

4.3.3 Implementations

A number of public-facing implementations of DP algorithms concern the generation of private, synthetic data on population commuting patterns using source data collected by the United States Census Bureau [173], the RAPPOR algorithm implemented in the Google Chrome browser [174], and the scalable local DP algorithms by Apple [175]. The geo-indistinguishability framework provides a principled approach for location privacy [176,177]. Finally, a number of open-source tools are available for users to apply well-known DP algorithms in the literature in their own data analytic tasks; we list them in Table 3 as references for the interested reader.

5 ROBUSTNESS

The concept of robustness in statistics and data-analytic exercises in general dates back decades ago to [178]. Broadly speaking, robust methods refer to techniques that are unaffected by the presence of outliers or other departures from model assumptions in the data used to implement them. Robust versions of many statistical methods have been proposed, and it is an active field of research—which will be discussed in another chapter in this book.

As more and more large-scale ML systems get deployed and updated in an automated manner, there is need for a different kind of robustness. This is robustness specifically against samples that are not 'similar' to the typical data a model was trained on. Oftentimes, such examples are tweaked by malicious actors (called as adversaries) in a targeted manner to make a ML model perform badly. Thus, this notion of robustness—hereafter called *adversarial robustness*—has a notion of security and reliability attached to it. Note that adversarial robustness is different from the traditional notion above. It concerns robustness with respect to perturbations in both the *test data* and training data, as compared to training data alone, and data contaminationations are generally tailored to the problem in hand.

In this section, we discuss the high-level concepts and research directions in adversarial robustness, with appropriate references as necessary. As in other sections, we refer interested readers to a number of survey articles for more details [179–181].

5.1 Adversarial attacks

Adversarial attacks on ML model have three categorizations—*who* they attack, *how* the attack happens, and *why* the attacks occur [180,181]. We consider only black-box models because of their ubiquitous nature in current ML, and since due to the transparency of white box model crafting targeted attacks are much easier. In the first category, *evasion attacks* occurs by maliciously adjusting testing samples, *data poisoning* attacks perturb the training data to corrupt the model training process, and *exploratory attacks* query a black-box model to reverse-engineer the training algorithm. As the second categorization, attacks can be orchestrated in a number of ways, depending on what information the attacker has access to. If an adversary has access to the training data or model, they can either modify the training data directly (data modification), add bad training data (data injection), or corrupt the trained model itself (logic corruption). On the other hand, in testing time adversaries can use the following kind of attacks on a black-box model:

1. **Adaptive attack:** Adversary labels a carefully constructed set of input feature through querying the target model, fits another ML model to predict these outputs, and tailors adversarial examples by focusing on areas where the second model has high error rates. Examples of adaptive attacks include [182,183].
2. **Non-adaptive attack:** Adversary has some prior knowledge about the training data distribution, which they use to generate the input data and predictions to fit a second-level model as above. Examples include [184,185].
3. **Strict attack:** Such attacks occur when the adversary uses actual input-output pairs from the original model to craft their attack. An example of strict attack is the method by [186].

Finally, as the third characterization, there are multiple possible goals of adversarial attacks. In decreasing order of specificity, they may aim for (a) general confidence reduction by deteriorating performance of the ML model as a whole, (b) misclassification of all input examples, (c) targeted misclassification of all input examples into specific classes, and (d) targeted misclassification of specific input examples into specific classes.

Not all attacks are equally difficult to perform, or equally effective. As a rule thumb, increasingly complex attacks are more difficult to perform, and tend to be more targeted to specific examples or tasks. We refer the reader to Figure 5 in [181] for a full comparison.

5.2 Defense mechanisms

We now discuss three broad classes of defense mechanisms against adversarial attacks: adversarial training (or retraining), regularization, and certified defenses. As somewhat expected, each of these strategies are effective against certain types of adversarial attacks, and come with their own performance guarantees.

5.2.1 Adversarial (re)training

Adversarial training is a popular method of adversarial defense, where the modeler wants to ensure robustness against certain types of adversaries during the model training phase. This can be done in a number of ways. Adversarial perturbations can be directly added to the training data, without any change to the training algorithm [187,188]. Methods that include defense measures incorporated into the learning model itself include minimizing the loss function over a grid of small perturbations around input points [189], ensemble training [190], training using an adversary critic [191], ME-Net [192], and misclassification-aware training [193].

5.2.2 Use of regularization

A number of adversarially robust optimization methods aim to limit the effect of small perturbations on input or outputs through controlling gradient updates during iterative model training. To this end, they use different kinds of norm bounds (such as ℓ_2, ℓ_∞) as additional constraints added to the overall loss function, or changing the gradient itself. Such algorithms include Parseval networks [194], DeepDefense [195], and TRADES [196]. We also refer the reader to Table 3 in [180] for a number of (re)training and regularization methods that use norm constraints.

5.2.3 Certified defenses

Compared to the above two, the strength of this class of methods is that they provide probabilistic guarantees for the robustness of the resulting model. One of the first works in this domain is Reluplex [197], which provides a baseline

framework for small neural networks with ReLU activation function. Subsequent research expanded this idea for more general networks [198] and other activation functions [199]. Recent research has also established theoretical guarantees for other quantities, such as a lower bound on the minimum necessary perturbation required to affect predictions of a model [200], and upper bounds on adversarial loss [201,202].

5.3 Implementations

We conclude with two implementations of adversarially robust algorithms that are available as open source packages. The Adversarial Robustness Toolbox (ART)[6] was released by IBM in 2019, and enables researchers and practitioners to evaluate and defend ML models evasion, poisoning, extraction and exploratory attacks [203]. The second package, AdvBox[7] was released by Baidu in January 2020 and provides similar functionalities. Both toolboxes are in Python. As the field of Adversarial ML matures with the increasing need for safe and reliable ML-based applications in the real world, efforts like these are essential towards faster adoption of latest research.

6 DISCUSSION

In this chapter, we have provided an overview of the technical methods for codifying human-centered values into large-scale ML systems. While there has been much research on this area in the last few years, a clear divide exists between theory and practice. Implementation frameworks of trustworthy ML methods are few and far between, and there are several challenges to develop and deploy such solutions in the wild [15,44,204,205]. To this end, it is critical to think about the cascading effects of big data collection processes and algorithmic systems built on such data that affect society at large—and vice versa. We sincerely hope that this chapter motivates practitioners and domain experts working on diverse application areas in leveraging their expertise to build participatory data-driven solutions that contribute towards the benefit of humankind.

BIBLIOGRAPHY

[1] J. Cook, Amazon scraps 'sexist AI' recruiting tool that showed bias against women (oct 2018).
URL https://www.telegraph.co.uk/technology/2018/10/10/amazon-scraps-sexist-ai-recruiting-tool-showed-bias-against/

[2] S. Perez, Facebook is removing over 5,000 ad targeting options to prevent discriminatory ads (aug 2018).
URL https://techcrunch.com/2018/08/21/facebook-is-removing-over-5000-

6. https://github.com/Trusted-AI/adversarial-robustness-toolbox
7. https://github.com/advboxes/AdvBox

ad-targeting-options-to-prevent-discriminatory-ads

[3] L. Cheng, K. R. Varshney, H. Liu, Socially responsible ai algorithms: Issues, purposes, and challenges, arXiv:2101.02032 (2021).

[4] E. Toreini, M. Aitken, K. Coopamootoo, et al., The Relationship between Trust in AI and Trustworthy Machine Learning Technologies, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, p. 272–283.

[5] P. Xiong, S. Buffett, S. Iqbal, et al., Towards a Robust and Trustworthy Machine Learning System Development, arXiv:2101.03042 (2021).

[6] S. Mitchell, E. Potash, S. Barocas, et al., Algorithmic Fairness: Choices, Assumptions, and Definitions, Annual Review of Statistics and Its Application 8 (1) (2021).

[7] N. Mehrabi, F. Morstatter, N. Saxena, et al., A Survey on Bias and Fairness in Machine Learning, arXiv:1908.09635 (2019).

[8] Y. Shrestha, Y. Yang, Fairness in Algorithmic Decision-Making: Applications in Multi-Winner Voting, Machine Learning, and Recommender Systems, Algorithms 12 (199) (2019) 1–28.

[9] S. Verma, J. Rubin, Fairness defnitions explained, in: FairWare '18: Proceedings of the International Workshop on Software Fairness, 2018.

[10] M. Hardt, E. Price, N. Srebro, Equality of Opportunity in Supervised Learning, in: Advances in Neural Information Processing Systems, Vol. 29, 2016, pp. 3315–3323.

[11] R. K. E. Bellamy, K. Dey, M. Hind, et al., AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, arXiv:1810.01943 (2018).

[12] M. Kusner, J. Loftus, C. Russell, R. Silva, Counterfactual Fairness, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 4066–4076.

[13] M. Kearns, S. Neel, A. Roth, Z. Wu, Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness, in: International Conference on Machine Learning, 2018, pp. 2569–2577.

[14] M. Kearns, S. Neel, A. Roth, Z. Wu, An empirical study of rich subgroup fairness for machine learning, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 100–109.

[15] K. Holstein, J. Vaughan, H. Daume, et al., Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019.

[16] S. Samadi, U. Tantipongpipat, J. H. Morgenstern, et al., The Price of Fair PCA: One Extra dimension, in: Advances in Neural Information Processing Systems, Vol. 31, 2018, pp. 10976–10987.

[17] A. Backurs, et al., Scalable Fair Clustering, in: International Conference on Machine Learning, Vol. 97, 2019, pp. 405–413.

[18] N. Mehrabi, F. Morstatter, N. Peng, A. Galstyan, Debiasing Community Detection: The Importance of Lowly Connected Nodes, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2019, p. 509–512.

[19] L. Zhang, Y. Wu, X. Wu, A Causal Framework for Discovering and Removing Direct and Indirect Discrimination, in: IJCAI-2017, 2017.

[20] M. Feldman, S. A. Friedler, J. Moeller, et al., Certifying and Removing Disparate Impact, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 259–268.

[21] R. Zemel, et al., Learning Fair Representations, in: Proceedings of the 30th International Conference on Machine Learning, Vol. 28, 2013, pp. 325–333.

[22] F. Calmon, et al., Optimized Pre-Processing for Discrimination Prevention, in: Advances in

Neural Information Processing Systems, Vol. 30, 2017, pp. 3992–4001.

[23] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowledge and Information Systems 33 (1) (2012) 1–33.

[24] B. Zhang, B. Lemoine, M. Mitchell, Mitigating Unwanted Biases with Adversarial Learning, in: AAAI Conference on AI, Ethics and Society, 2018.

[25] T. Kamishima, et al., Fairness-Aware Classifier with Prejudice Remover Regularizer, in: Machine Learning and Knowledge Discovery in Databases, 2012, pp. 35–50.

[26] L. Celis, et al., Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2018, pp. 319–328.

[27] F. Kamiran, A. Karim, X. Zhang, Decision Theory for Discrimination-Aware Classification, in: IEEE 12th International Conference on Data Mining, 2012, pp. 924–929.

[28] G. Pleiss, et al., On Fairness and Calibration, in: Advances in Neural Information Processing Systems, 2017, pp. 5680–5689.

[29] A. Stevens, et al., Aequitas: Bias and fairness audit, Tech. rep., Center for Data Science and Public Policy, The University of Chicago, `https://github.com/dssg/aequitas` (2018).

[30] M. Zehlike, et al., Fairness Measures: Datasets and software for detecting algorithmic discrimination, `http://fairness-measures.org` (2017).

[31] J. A. Adebayo, FairML: Toolbox for diagnosing bias in predictive modeling, Master's thesis, MIT, `https://github.com/adebayoj/fairml` (2016).

[32] A. Tramer, et al., FairTest: Discovering unwarranted associations in data-driven applications, in: EuroS&P-2017, 2017.

[33] S. Galhotra, Y. Brun, A. Meliou, Fairness Testing: Testing software for discrimination, in: ESEC/FSE-2017, 2017.

[34] M. Dudik, et al., Fairlearn, `https://github.com/fairlearn/fairlearn` (2020).

[35] N. Bantilan, Themis-ml: A Fairness-Aware ML Interface for End-To-End Discrimination Discovery and Mitigation, Journal of Technology in Human Services 36 (1) (2018) 15–30.

[36] M. Veale, M. V. Cleek, R. Binns, Fairness and accountability design needs for algorithmic support in highstakes public sector decision-making, in: CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018.

[37] Gebru, T. and J. Morgenstern and B. Vecchione and others, Datasheets for datasets, arXiv:1908.09635 (2018).

[38] S. Holland, A. Hosny, S. Newman, et al., The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards, arXiv:arxiv.org/abs/1805.03677 (2018).

[39] M. Arnold, R. Bellamy, M. Hind, et al., FactSheets: Increasing Trust in AI Services through Supplier's Declarations of Conformity, IBM Journal of Research and Development 63 (4/5) (2019) 6:1–6:13.

[40] M. Mitchell, S. Wu, A. Zaldivar, et al., Model Cards for Model Reporting, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 220–229.

[41] S. Vasudevan, K. Kenthapadi, LiFT: A Scalable Framework for Measuring Fairness in ML Applications, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020.

[42] I. Raji, A. Smart, R. White, et al., Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019.

[43] M. Madaio, L. Stark, J. Vaughan, H. Wallach, Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020.

[44] E. Dodwell, C. Flynn, B. Krishnamurthy, et al., Towards Integrating Fairness Transparently in Industrial Applications, arXiv:2006.06082 (2020).

[45] M. Du, N. Liu, X. Hu, Techniques for Interpretable Machine Learning, Communications of the ACM 63 (1) (2020) 68–77.

[46] D. V. Carvalho, E. M. Pereira, J. S. Cardoso, Machine Learning Interpretability: A Survey on Methods and Metrics, Electronics 8 (832) (2019) 1–34.

[47] A. Adadi, M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access 6 (2018) 52138–52160.

[48] L. H. Gilpin, D. Bau, B. Z. Yuan, et al., Explaining Explanations: An Overview of Interpretability of Machine Learning, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 2018, pp. 80–89.

[49] A. Holzinger, G. Langs, H. Denk, et al., Causability and explainability of artificial intelligence in medicine, WIREs Data Mining Knowledge Discovery 9 (e1312) (2019) 1–13.

[50] I. C. Covert, S. Lundberg, S.-I. Lee, Feature Removal Is A Unifying Principle For Model Explanation Methods, in: NeurIPS 2020 ML-Retrospectives, Surveys & Meta-Analyses Workshop, 2020, arXiv:2011.03623.

[51] Z. C. Lipton, The Mythos of Model Interpretability, in: 2016 ICML Workshop on Human Interpretability in Machine Learning, 2016, arXiv:1606.03490.

[52] F. Doshi-Velez, B. Kim, Towards A Rigorous Science of Interpretable Machine Learning, arXiv:1702.08608 (2017).

[53] S. Bromburger, On what we know we don't know: Explanation, theory, linguistics, and how questions shape them, University of Chicago Press, 1992.

[54] B. Herman, The Promise and Peril of Human Evaluation for Model Interpretability, in: Proceedings of NIPS 2017 Symposium on Interpretable Machine Learning, 2017, arXiv:1711.07414.

[55] R. Moraffah, M. Karami, R. Guo, et al., Causal Interpretability for Machine Learning - Problems, Methods and Evaluation, ACM SIGKDD Explorations Newsletter 22 (1) (2020) 18–33.

[56] J. Pearl, Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018.

[57] M. T. Ribeiro, S. Singh, C. Guestrin, Ẅhy Should I Trust You?̈: Explaining the Predictions of Any Classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[58] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4768–4777.

[59] H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Faithful and Customizable Explanations of Black Box Models, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 131–138.

[60] A. Fisher, C. Rudin, F. Dominici, All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously, Journal of Machine Learning Research 20 (177) (2019) 1–81.

[61] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (2019) 206–215.

[62] Y. Lou, R. Caruana, J. Gehrke, Intelligible models for classification and regression, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 150–158.

[63] R. Guo, L. Cheng, J. Li, et al., A Survey of Learning Causality with Data: Problems and Methods, ACM Computing Surveys 53 (4) (2020).

[64] J. Pearl, Causality: Models, Reasoning and Inference, Cambridge University Press, 2000.

[65] M. Harradon, J. Druce, B. Ruttenberg, Causal Learning and Explanation of Deep Neural Networks via Autoencoded Activations, arXiv:1802.00541 (2018).

[66] T. Narendra, A. Sankaran, D. Vijaykeerthy, S. Mani, Explaining deep learning models using causal inference, arXiv:1811.04376 (2018).

[67] A. Chattopadhyay, P. Manupriya, A. Sarkar, V. N. Balasubramanian, Neural network attributions: A causal perspective, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, 2019, pp. 981–990.

[68] Q. Zhao, T. Hastie, Causal interpretations of black-box models, Journal of Business & Economic Statistics 39 (1) (2021) 272–281.

[69] A. Parafita, J. Vitriá, Explaining visual models by causal attribution, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 4167–4175.

[70] P. Madumal, T. Miller, L. Sonenberg, F. Vetere, Explainable reinforcement learning through a causal lens, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 2493–2500.

[71] S. Wachter, B. D. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, arXiv:1711.00399 (2017).

[72] Y. Goyal, U. Shalit, B. Kim, Explaining classifiers with causal concept effect (CaCE), arXiv:1907.07165 (2019).

[73] J. Moore, N. Hammerla, C. Watkins, Explaining deep learning models with constrained adversarial examples, in: PRICAI 2019: Trends in Artificial Intelligence, 2019, pp. 43–56.

[74] R. K. Mothilal, A. Sharma, C. Tan, Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, p. 607–617.

[75] S. Rathi, Generating Counterfactual and Contrastive Explanations using SHAP, arXiv:1906.09293 (2019).

[76] L. A. Hendricks, R. Hu, T. Darrell, Z. Akata, Generating Counterfactual Explanations with Natural Language, arXiv:1806.09809 (2018).

[77] N. Kilbertus, M. R. Carulla, G. Parascandolo, et al., Avoiding discrimination through causal reasoning, in: Advances in Neural Information Processing Systems 30, 2017, pp. 656–666.

[78] D. Madras, E. Creager, T. Pitassi, R. Zemel, Fairness through Causal Awareness: Learning Causal Latent-Variable Models for Biased Data, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, p. 349–358.

[79] J. Zhang, E. Bareinboim, Fairness in Decision-Making—the Causal Explanation Formula, in: AAAI-2018, 2018.

[80] C. Kim, O. Bastani, Learning Interpretable Models with Causal Guarantees, arXiv:1901.08576 (2019).

[81] R. Caruana, Y. Lou, J. Gehrke, et al., Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1721–1730.

[82] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38.

[83] T. Miller, P. Howe, , L. Sonenberg, Explainable AI: Beware of inmates running the asylum, in: Proceedings of IJCAI Workshop Explainable AI (XAI), 2017, pp. 36–42.

[84] B. Kim, R. Khanna, O. O. Koyejo, Examples are not enough, learn to criticize! Criticism for Interpretability, in: Advances in Neural Information Processing Systems, Vol. 29, 2016, pp.

2280–2288.

[85] K. S. Gurumoorthy, A. Dhurandhar, G. A. Cecchi, C. C. Aggarwal, Efficient Data Representation by Selecting Prototypes with Importance Weights, in: 2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019, 2019, pp. 260–269.

[86] J. Wortman Vaughan, H. Wallach, A Human-Centered Agenda for Intelligible Machine Learning, http://www.jennwv.com/papers/intel-chapter.pdf (2020).

[87] M. Bauer, S. Baldes, An Ontology-Based Interface for Machine Learning, in: Proceedings of the 10th International Conference on Intelligent User Interfaces, 2005, p. 314–316.

[88] J. Zhu, A. Liapis, S. Risi, et al., Explainable AI for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation, in: 2018 IEEE Conference on Computational Intelligence and Games (CIG), 2018, pp. 1–8.

[89] P. Tamagnini, J. Krause, A. Dasgupta, E. Bertini, Interpreting Black-Box Classifiers Using Instance-Level Visual Explanations, in: Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics, 2017.

[90] F. Hohman, A. Head, R. Caruana, et al., Gamut: A Design Probe to Understand How Data Scientists Understand Machine Learning Models, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, p. 1–13.

[91] D. Alvarez-Melis, H. Daumé, J. W. Vaughan, H. Wallach, Weight of Evidence as a Basis for Human-Oriented Explanations, in: Human-Centric Machine Learning (HCML) Workshop @ NeurIPS 2019, 2019, arXiv:1910.13503.

[92] R. Barraza, R. Eames, Y. E. Balducci, et al., Error terrain analysis for machine learning: Tool and visualizations, in: ICLR workshop on debugging machine learning models, 2019.

[93] S. Amershi, M. Chickering, S. M. Drucker, et al., ModelTracker: Redesigning performance analysis tools for machine learning, in: Proceedings of the 2015 CHI conference on human factors in computing systems (CHI), 2015, pp. 337–346.

[94] D. Ren, S. Amershi, B. Lee, et al., Squares: Supporting interactive performance analysis for multiclass classifiers, IEEE Transactions on Visualization and Computer Graphics 23 (1) (2016) 61–70.

[95] H. Kaur, H. Nori, S. Jenkins, et al., Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, no. 92, 2020, p. 1–14.

[96] H. Lakkaraju, O. Bastani, "How Do I Fool You?¨: Manipulating User Trust via Misleading Black Box Explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, 2020, p. 79–85.

[97] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, et al., Manipulating and Measuring Model Interpretability , in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, arXiv:1802.07810.

[98] V. Arya, R. K. E. Bellamy, P.-Y. Chen, et al., One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques, arXiv:1909.03012 (2019).

[99] H. Nori, S. Jenkins, P. Koch, R. Caruana, InterpretML: A Unified Framework for Machine Learning Interpretability, arXiv:1909.09223 (2019).

[100] S. Maksymiuk, A. Gosiewska, P. Biecek, Landscape of R packages for eXplainable Artificial Intelligence, arXiv:2009.13248 (2020).

[101] S. Vadhan, The Complexity of Differential Privacy, in: Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich, Springer International Publishing, 2017, pp. 347–450.

[102] A. Narayanan, J. Huey, E. W. Felten, A Precautionary Approach to Big Data Privacy, in:

Data Protection on the Move: Current Developments in ICT and Privacy/Data Protection, Springer Netherlands, 2016, pp. 357–385.

[103] F. du Pin Calmon, N. Fawaz, Privacy against statistical inference, in: 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2012, pp. 1401–1408.

[104] C. Dwork, A. Roth, The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science 9 (3–4) (2014) 211–407.

[105] G. Kamath, J. Ullman, A Primer on Private Statistics, arXiv:2005.00010 (2020).

[106] A. Wood, M. Altman, A. Bembenek, et al., Differential privacy: A primer for a non-technical audience, Vanderbilt Journal of Entertainment & Technology Law 21 (1) (2018) 209–275.

[107] M. Gong, Y. Xie, K. Pan, et al., A Survey on Differentially Private Machine Learning [Review Article], IEEE Computational Intelligence Magazine 15 (2) (2020) 49–64.

[108] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating Noise to Sensitivity in Private Data Analysis, in: Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 265–284.

[109] A. Blum, C. Dwork, F. McSherry, K. Nissim, Practical Privacy: The SuLQ Framework, in: Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2005, p. 128–138.

[110] F. McSherry, K. Talwar, Mechanism Design via Differential Privacy, in: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), 2007, pp. 94–103.

[111] F. D. McSherry, Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, 2009, p. 19–30.

[112] K. Nissim, S. Raskhodnikova, A. Smith, Smooth Sensitivity and Sampling in Private Data Analysis, in: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, 2007, p. 75–84.

[113] V. A. E. Farias, F. T. Brito, C. Flynn, et al., Local Dampening: Differential Privacy for Non-numeric Queries via Local Sensitivity, Proceedings of the VLDB Endowment 14 (4) (2020) 521 – 533.

[114] A. Gonem, R. Gilad-Bachrach, Smooth Sensitivity Based Approach for Differentially Private PCA, in: Proceedings of Algorithmic Learning Theory, 2018, pp. 438–450.

[115] V. Karwa, S. Raskhodnikova, A. Smith, G. Yaroslavtsev, Private Analysis of Graph Structure, ACM Transactions on Database Systems 39 (3) (2014).

[116] L. Sun, Y. Zhou, P. S. Yu, C. Xiong, Differentially Private Deep Learning with Smooth Sensitivity, arXiv:2003.00505 (2020).

[117] M. Bun, T. Steinke, Average-Case Averages: Private Algorithms for Smooth Sensitivity and Mean Estimation, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 181–191.

[118] J. Zhang, G. Cormode, C. M. Procopiuc, et al., Private Release of Graph Statistics Using Ladder Functions, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015, p. 731–745.

[119] Q. Geng, P. Kairouz, S. Oh, P. Viswanath, The staircase mechanism in differential privacy, IEEE Journal of Selected Topics in Signal Processing 9 (7) (2015) 1176–1184.

[120] R. McKenna, D. R. Sheldon, Permute-and-Flip: A new mechanism for differentially private selection , in: Advances in Neural Information Processing Systems, 2020.

[121] X. Liu, Q. Li, T. Li, D. Chen, Differentially private classification with decision tree ensemble, Applied Soft Computing 62 (2018) 807 – 816.

[122] J. Vaidya, B. Shafiq, A. Basu, Y. Hong, Differentially Private Naive Bayes Classification,

in: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Vol. 1, 2013, pp. 571–576.

[123] T. Li, J. Li, Z. Liu, et al., Differentially private Naive Bayes learning over multiple data sources, Information Sciences 444 (2018) 89 – 104.

[124] C. Dwork, G. N. Rothblum, S. Vadhan, Boosting and Differential Privacy, in: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 51–60.

[125] G. Jagannathan, K. Pillaipakkamnatt, R. N. Wright, A Practical Differentially Private Random Decision Tree Classifier, Trans. Data Privacy 5 (1) (2012) 273–295.

[126] S. Rana, S. K. Gupta, S. Venkatesh, Differentially Private Random Forest with High Utility, in: 2015 IEEE International Conference on Data Mining, 2015, pp. 955–960.

[127] S. Fletcher, M. Zahidul Islam, A Differentially Private Decision Forest, in: Proceedings of the 13-th Australasian Data Mining Conference (AusDM 2015), 2015, pp. 99–108.

[128] D. Su, J. Cao, N. Li, et al., Differentially private k-means clustering, in: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, 2016, p. 26–37.

[129] V. Schellekens, A. Chatalic, F. Houssiau, et al., Differentially private compressive k-means, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 7933–7937.

[130] K. Chaudhuri, A. D. Sarwate, K. Sinha, A Near-Optimal Algorithm for Differentially-Private Principal Components, Journal of Machine Learning Research 14 (2013) 2905–2943.

[131] P. Jain, P. Kothari, A. Thakurta, Differentially Private Online Learning, in: Proceedings of the 25th Annual Conference on Learning Theory, 2012, pp. 24.1–24.34.

[132] C. Li, P. Zhou, L. Xiong, Q. Wang, T. Wang, Differentially private distributed online learning, IEEE Transactions on Knowledge and Data Engineering 30 (8) (2018) 1440–1453.

[133] S. Yildirim, B. Ermis, Exact MCMC with differentially private moves, Stat Comput 29 (2019) 947–963.

[134] Mikko A. Heikkilä and Joonas Jälkö and Onur Dikmen and Antti Honkela, Differentially Private Markov Chain Monte Carlo, in: NeurIPS-2019, 2019.

[135] L. Lode, Sub-sampled and Differentially Private Hamiltonian Monte Carlo, Master's thesis, University of Helsinki (2019).

[136] M. Park, J. Foulds, K. Choudhary, M. Welling, DP-EM: Differentially Private Expectation Maximization, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Vol. 54, 2017, pp. 896–904.

[137] S. Song, K. Chaudhuri, A. D. Sarwate, Stochastic gradient descent with differentially private updates, in: 2013 IEEE Global Conference on Signal and Information Processing, 2013, pp. 245–248.

[138] A. Rajkumar, S. Agarwal, A Differentially Private Stochastic Gradient Descent Algorithm for Multiparty Classification, in: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, 2012, pp. 933–941.

[139] M. Abadi, A. Chu, I. Goodfellow, et al., Deep Learning with Differential Privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, p. 308–318.

[140] J. Zhang, Z. Zhang, X. Xiao, et al., Functional Mechanism: Regression Analysis under Differential Privacy, Proceedings of the VLDB Endowment 5 (11) (2012) 1364–1375.

[141] J. Lei, Differentially Private M-Estimators, in: Advances in Neural Information Processing Systems 24, 2011, pp. 361–369.

[142] K. Talwar, A. Thakurta, L. Zhang, Nearly-Optimal Private LASSO, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, 2015, p. 3025–3033.

[143] Y. Zhang, Z. Hao, S. Wang, A Differential Privacy Support Vector Machine Classifier Based on Dual Variable Perturbation, IEEE Access 7 (2019) 98238–98251.

[144] P. Jain, A. Thakurta, Differentially private learning with kernels, in: Proceedings of the 30th International Conference on Machine Learning, Vol. 28, 2013, pp. 118–126.

[145] K. Chaudhuri, A. D. Sarwate, K. Sinha, Differentially Private Empirical Risk Minimization, Journal of Machine Learning Research 12 (2011) 1069–1109.

[146] D. Kifer, A. Smith, A. Thakurta, Private convex empirical risk minimization and high-dimensional regression, in: Proceedings of the 25th Annual Conference on Learning Theory, Vol. 23, 2012, pp. 25.1–25.40.

[147] D. Wang, C. Chen, J. Xu, Differentially Private Empirical Risk Minimization with Non-convex Loss Functions, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, 2019, pp. 6526–6535.

[148] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2015, pp. 909–910.

[149] X. Zhang, S. Ji, H. Wang, T. Wang, Private, yet practical, multiparty deep learning, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 1442–1452.

[150] N. Papernot, S. Song, I. Mironov, et al., Scalable Private Learning with PATE, in: ICLR 2018, 2018.

[151] H. B. McMahan, D. Ramage, K. Talwar, L. Zhang, Learning Differentially Private Recurrent Language Models (2018).

[152] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, et al., Privacy-Preserving Generative Deep Neural Networks Support Clinical Data Sharing, Circulation: Cardiovascular Quality and Outcomes 12 (7) (2019) e005122.

[153] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially Private Generative Adversarial Network, arXiv:1802.06739 (2018).

[154] N. Phan, Y. Wang, X. Wu, D. Dou, Differential Privacy Preservation for Deep Auto-Encoders: An Application of Human Behavior Prediction, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, p. 1309–1316.

[155] N. Phan, X. Wu, D. Dou, Preserving Differential Privacy in Convolutional Deep Belief Networks 106 (9–10) (2017) 1681–1704.

[156] D. Xu, S. Yuan, X. Wu, et al., DPNE: Differentially Private Network Embedding, in: Advances in Knowledge Discovery and Data Mining, 2018, pp. 235–246.

[157] D. Kifer, A. Machanavajjhala, Pufferfish: A framework for mathematical privacy definitions, ACM Transactions on Database Systems 39 (1) (2014).

[158] S. Song, Y. Wang, K. Chaudhuri, Pufferfish Privacy Mechanisms for Correlated Data, in: Proceedings of the 2017 ACM International Conference on Management of Data, 2017, p. 1291–1306.

[159] L. Ou, Z. Qin, S. Liao, et al., An Optimal Pufferfish Privacy Mechanism for Temporally Correlated Trajectories, IEEE Access 6 (2018) 37150–37165.

[160] W. Liang, H. Chen, R. Liu, et al., A Pufferfish privacy mechanism for monitoring web browsing behavior under temporal correlations, Computers & Security 92 (2020) 101754.

[161] Z. Xi, Y. Sang, H. Zhong, et al., Pufferfish Privacy Mechanism Based on Multi-dimensional Markov Chain Model for Correlated Categorical Data Sequences, in: Parallel Architectures, Algorithms and Programming, Springer Singapore, Singapore, 2020, pp. 430–439.

[162] X. He, A. Machanavajjhala, B. Ding, Blowfish Privacy: Tuning Privacy-Utility Trade-Offs Using Policies, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 2014, p. 1447–1458.

[163] R. Chen, B. C. Fung, P. S. Yu, B. C. Desai, Correlated Network Data Publication via Differential Privacy, The VLDB Journal 23 (4) (2014) 653–676.

[164] Y. Cao, M. Yoshikawa, Y. Xiao, L. Xiong, Quantifying Differential Privacy under Temporal Correlations, in: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), 2017, pp. 821–832.

[165] B. Palanisamy, C. Li, P. Krishnamurthy, Group Differential Privacy-Preserving Disclosure of Multi-level Association Graphs, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 2587–2588.

[166] I. Mironov, Rényi Differential Privacy, in: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), 2017, pp. 263–275.

[167] C. Dwork, G. N. Rothblum, Concentrated Differential Privacy, arXiv:1603.01887 (2016).

[168] M. Bun, T. Steinke, Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds, in: Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 635–658.

[169] K. Chaudhuri, J. Imola, A. Machanavajjhala, Capacity Bounded Differential Privacy, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 3474–3483.

[170] Y. Zhu, Y.-X. Wang, Poission Subsampled Rényi Differential Privacy, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 7634–7642.

[171] C. M. O'Keefe, A.-S. Charest, Bootstrap Differential Privacy, Transactions on Data Privacy 12 (2019) 1–28.

[172] J. Dong, A. Roth, W. J. Su, Gaussian Differential Privacy, Journal of the Royal Statistical Society: Series B arXiv: 1905.02383 (2021).

[173] A. Machanavajjhala, D. Kifer, J. Abowd, et al., Privacy: Theory meets practice on the map, in: 2008 IEEE 24th International Conference on Data Engineering, 2008, pp. 277–286.

[174] Ú. Erlingsson, V. Pihur, A. Korolova, RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014, 2014, pp. 1054–1067.

[175] Differential Privacy Team, Apple, Learning with Privacy at Scale, available at: `https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf` (2017).

[176] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geo-Indistinguishability: Differential Privacy for Location-Based Systems, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, p. 901–914.

[177] S. Oya, C. Troncoso, F. Pérez-González, Is Geo-Indistinguishability What You Are Looking For?, in: Proceedings of the 2017 on Workshop on Privacy in the Electronic Society, 2017, p. 137–140.

[178] P. J. Huber, Robust statistics, John Wiley & Sons, Inc., New York, 1981.

[179] A. Serban, E. Poll, J. Visser, Adversarial Examples on Object Recognition: A Comprehensive Survey, ACM Computing Surveys 53 (3) (2020).

[180] S. H. Silva, P. Najafirad, Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey, arXiv:2007.00753 (2020).

[181] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, Adversarial attacks and defences: A survey, arXiv:1810.00069 (2018).

[182] M. Fredrikson, S. Jha, T. Ristenpart, Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, p. 1322–1333.

[183] I. Rosenberg, A. Shabtai, L. Rokach, Y. Elovici, Generic Black-Box End-to-End Attack

Against State of the Art API Call Based Malware Classifiers, in: Research in Attacks, Intrusions, and Defenses, Springer International Publishing, 2018, pp. 490–510.

[184] F. Tramèr, F. Zhang, A. Juels, et al., Stealing Machine Learning Models via Prediction APIs, in: Proceedings of the 25th USENIX Conference on Security Symposium, 2016, p. 601–618.

[185] N. Papernot, P. McDaniel, I. Goodfellow, Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, arXiv:1605.07277 (2016).

[186] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, p. 603–618.

[187] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and Harnessing Adversarial Examples, in: International Conference on Learning Representations, 2015.

[188] E. Wong, L. Rice, J. Z. Kolter, Fast is better than free: Revisiting adversarial training, in: International Conference on Learning Representations, 2019.

[189] A. Madry, A. Makelov, L. Schmidt, et al., Towards Deep Learning Models Resistant to Adversarial Attacks, in: International Conference on Learning Representations, 2018.

[190] F. Tramer, A. Kurakin, N. Papernot, et al., Ensemble Adversarial Training: Attacks and Defenses, in: International Conference on Learning Representations, 2018.

[191] A. Matyasko, L.-P. Chau, Improved Network Robustness with Adversary Critic, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, p. 10601–10610.

[192] Y. Yang, G. Zhang, D. Katabi, Z. Xu, ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, 2019, pp. 7025–7034.

[193] Y. Wang, D. Zou, J. Yi, et al., Improving Adversarial Robustness Requires Revisiting Misclassified Examples, in: International Conference on Learning Representations, 2020.

[194] M. Cisse, P. Bojanowski, E. Grave, et al., Parseval Networks: Improving Robustness to Adversarial Examples, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, 2017, p. 854–863.

[195] Z. Yan, Y. Guo, C. Zhang, Deep Defense: Training DNNs with Improved Adversarial Robustness, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, p. 417–426.

[196] H. Zhang, Y. Yu, J. Jiao, et al., Theoretically Principled Trade-off between Robustness and Accuracy, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 7472–7482.

[197] G. Katz, C. Barrett, D. L. Dill, et al., Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, in: Computer Aided Verification, Springer International Publishing, 2017, pp. 97–117.

[198] T. Gehr, M. Mirman, D. Drachsler-Cohen, et al., AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation, in: 2018 IEEE Symposium on Security and Privacy (SP), 2018, pp. 3–18.

[199] G. Singh, T. Gehr, M. Mirman, et al., Fast and Effective Robustness Certification, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, p. 10825–10836.

[200] M. Hein, M. Andriushchenko, Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 2266–2276.

[201] A. Raghunathan, J. Steinhardt, P. Liang, Certified Defenses against Adversarial Examples, in: International Conference on Learning Representations, 2018.

[202] E. Wong, J. Z. Kolter, Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope, in: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, 2018, pp. 5283–5292.

[203] B. Buesser, A. Goldsteen, Adversarial Robustness Toolbox: One Year Later with v1.4, https://www.ibm.com/blogs/research/2020/10/adversarial-robustness-toolbox-one-year-later-with-v1-4.

[204] A. Brennen, What Do People Really Want When They Say They Want "Explainable AI?" We Asked 60 Stakeholders, in: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, 2020.

[205] M. Andrus, E. Spitzer, J. Brown, A. Xiang, "What We Can't Measure, We Can't Understand": Challenges to Demographic Data Procurement in the Pursuit of Fairness, in: Proceedings of the 2021 Conference on Fairness, Accountability, and Transparency, 2021.