Thesis Proposal
Accountable Information Use in Data-Driven Systems

Shayak Sen

May 1, 2017

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Anupam Datta, Chair
Jaime Carbonell
Matt Fredrikson
Sriram K. Rajamani
Jeannette M. Wing

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Contents

Chapter 1

Introduction

Data-driven systems span a wide class of application domains, many of which have a signficant impact on people's lives. Examples of such domains include credit, insurance, predictive policing, and personalized advertising. Fueled by the increased and centralized availability of personal information, along with algorithmic and computational advances, such systems have become fairly straightforward to develop and deploy. However, due to their size and complexity, many of these systems are largely opaque; automated loan denials, high interest rates, and prison sentence recommendations are not usually accompanied by any explanation of how such decisions are made. As a result, the impact these systems bear on people's lives, combined with how opaque these systems are, has led to a strong call for accountability for these systems from various quarters[7, 10, 25, 40] in order to address threats to privacy and fairness in the operation of these systems. We use the term "accountable" to refer to computational mechanisms that can be used to support detection of privacy and fairness violations, as well as explain how they came about. We then leverage this understanding to repair systems to avoid future violations.

In this thesis, we will focus on privacy and fairness harms that arise out of improper information use. Limitations on information use are already well recognized norms in the domains described above. For example, the *use limitation* norms in law and guidelines such as the FTC's FIPPs in United States [87], the PIPEDA in Canada [72], and the GDPR in the European Union [39], require information use to only be limited to the purposes for which it was collected, and additionally restrict the use of sensitive information types such as health status, and sexual orientation. Anti-discrimination laws in employment [4], housing, credit [5] prevent the use of protected attributes such as gender, race, nationality, and sexual orientation for making decisions.

However, the enforcement of limitations on information use in data-driven systems presents some significant challenges.

Scale. In large codebases maintained by multiple parties, where information may be used for many purposes, identifying prohibited uses is a challenging task. Further, in settings where policies are framed and interpreted by privacy professionals independently from the code developers, compliance workflows are often manual and therefore have low coverage.

Opacity. Systems that use machine learning are particularly opaque, even when their programs are available for inspection. In the absence of an explanation for behaviors, determining compliance is a non-trivial task.

Proxies. Information may be used indirectly through proxies even if it isn't directly provided to the system. Accounting for information use through proxies, and eliminating them entails accounting for associations that may be present between features and sensitive attributes.

Normative Considerations. Privacy and non-discrimination constraints often contain exceptions based on normative ethical considerations and therefore theories of information use need to be accompanied by mechanisms to express and allow such exceptions.

In this report, we describe completed and proposed work on analyzing information use that address these challenges in support of the following thesis.

> Tools for analyzing information use enable practical accountability mechanisms that ensure data-driven systems respect meaningful privacy and non-discrimination properties.

We distinguish between two forms of information use: explicit and proxy use. A system exhibits *explicit use* of an input if the input has a *causal effect* on the behavior of the system, that is, changing the input, while keeping other inputs fixed, changes the behavior of the system. The notion of explicit use is identical to that of interference in programs. This connection between use, causality, and interference has been made formal in [96].

However, information may also be used indirectly via inferences made from other data, even if the particular piece of information has not been explicitly provided to the system. We call such use *proxy use*. For example, the process of targeting residents of certain areas based on the composition of race or nationality of origin of that area, is an illegal practice known as economic redlining. In this case, area of residence can be used as a proxy for race or nationality of origin.

We now summarize the key completed and proposed works in support of this thesis, organized along these two forms of information use.

1.1 Explicit Use

A system exhibits *explicit use* of an input if the input has a *causal effect* on the behavior of the system, that is, changing the input, while keeping other inputs fixed, changes the behavior of the system. This notion of explicit use is identical to that of interference in programs. Verifying the absence of interference in programs has been the subject of much research, starting from the work of Denning [30]. Applying such techniques to a privacy compliance workflow for industrial-scale applications is a challenge we addressed in [83].

1.1.1 Completed: Bootstrapping Privacy Compliance at Scale

To contextualize the challenges in performing automated privacy compliance checking in a large company with tens of thousands of employees, it is useful to understand the division of labor and responsibilities in current compliance workflows [22, 54]. Privacy policies are typically crafted by lawyers in a corporate legal team to adhere to all applicable laws and regulations worldwide. Due to the rapid change in product features and internal processes, these policies are necessarily specified using high-level policy concepts that may not cleanly map to the products that are expected to comply with them. For instance, a policy may refer to “IP Address” which is a high-level policy concept, and the product may have thousands of data stores where data derived from the “IP Address” is stored (and called with different names) and several thousand processes that produce and consume this data, all of which have to comply with policy. The task of interpreting the policy as applicable to individual products then falls to the tens of privacy champions embedded in product groups. Privacy champions review product features at various stages of the development process, offering specific requirements to the development teams to ensure compliance with policy. The code produced by the development team is expected to adhere to these requirements. Periodically, the compliance team audits development teams to ensure that the requirements are met.

Our central contribution in this work is a workflow for privacy compliance in big data systems. Specifically, we target privacy compliance of large codebases written in languages that support the Map-Reduce programming model [17, 65, 88]. This focus enables us to apply our workflow to current industrial-scale data processing applications, in particular the data analytics backend of Bing, Microsoft’s web search engine [1]. This workflow leverages our two key technical contributions: (1) a language LEGALEASE for stating privacy policies, which is usable by policy authors and privacy policy champions, but has precise semantics and enables automated checking for compliance, and (2) a self-bootstrapping data inventory mapper GROK, which maps low level data types in code to high-level policy concepts, and bridges the world of product development with the world of policy makers, . These two contributions, are important components of a privacy compliance workflow, currently used by Bing’s data analytics backend.

1.1.2 Completed: Explaining and Quantifying Explicit Use

Systems that employ machine learned models in their data analytics pipeline pose a significant challenge to reasoning about how information is used due to their complexity. Many inputs are used as features, and as a result have some causal effect on outcomes, which may be very low. In [27], we develop a family of measures to quantify the causal influence of inputs of systems on their outcomes. These measures provide a foundation for the design of explanations that accompany system decisions (e.g., explaining a specific credit decision) and for testing tools useful for internal and external oversight (e.g., to detect algorithmic discrimination). Distinctively, our causal QII measures carefully account for correlated inputs while measuring influence. They support a general class of transparency queries and can, in particular, explain decisions about individuals (e.g., a loan decision) and groups (e.g., disparate impact based on gender). Finally, since single inputs may not always have high influence, the QII measures also quantify the joint influence of a set of inputs (e.g., age and income) on outcomes (e.g. loan decisions) and the marginal influence of individual inputs within such a set (e.g., income). Since a single input may be part of multiple influential sets, the average marginal influence of the input is computed using principled aggregation measures, such as the Shapley value, previously applied to measure influence in voting.

1.1.3 Proposed: Distributional Faithfulness.

In order to measure the causal influence of inputs, we observe the outcome of the classifier on counterfactual points, i.e. points which change one input while keeping all other inputs fixed. Such causal experimentation is the staple of much of the natural sciences. However, in a problem peculiar to machine learning models, since counterfactual inputs may lie outside of distribution the model was trained on, the model is not required to behave meaningfully on such inputs. We will take two approaches to address this problem. First, we will consider influence measures that restrict the set of counterfactuals to belong to the distribution that model was trained on, resulting in a *distribution faithful* influence measure. Second, we will explore methods to retrain the model on counterfactual points, in order bring the training distribution closer to the distribution of counterfactual points. We term this approach *counterfactual active learning*.

1.2 Proxy Use

While the first part of this work focuses on explicit use, information can be used indirectly through proxies, which allow a data processor to effectively infer protected information types and use them even when they are not explicitly provided. In ongoing work, we propose a theory of proxy use, and use it as a building block to construct theories of *use privacy* and *proxy non-discrimination*. Importantly, this focus on use is a significant departure from a large body of prior work that focuses on limiting disclosures for privacy (see [86] for a survey), and disparate impact for fairness [44, 56, 74, 91, 102], which can both be viewed forms of probabilistic association.

1.2.1 Ongoing Work: Use Privacy

Use privacy constraints restrict the use of protected information types and some of their proxies in data-driven systems.

A use privacy constraint may require that health information or its proxies not be used for advertising. Indeed, there are calls for this form of privacy constraint [29, 64, 70, 98]. In this work, we consider the setting where a data-driven system is audited to ensure that it complies with such use privacy constraints. The auditing could be done by a co-operative data processor who is operating the system or by a regulatory oversight organization who has access to the data processors' machine learning models and knowledge of the distribution of the dataset. In other words, we assume that the data processor does not act to evade the detection algorithm, and provides accurate information.

In this setting, it is impossible to guarantee that data processors with strong background knowledge are not able to infer certain facts about individuals [35]. Even in practice, data processors often have access to detailed profiles of individuals and can infer sensitive information about them [33, 97]. Use privacy instead places a more pragmatic requirement on data-driven systems: that they simulate ignorance of protected information types by not using them or their proxies in their decision-making. This requirement is met if the systems (e.g., machine learning models) do not infer protected information types or their proxies (even if they could) or if such inferences do not affect decisions.

Recognizing that not all instances of proxy use of a protected information type are inappropriate, our theory of use privacy makes use of a normative judgment oracle that makes this inappropriateness determination for a given instance. For example, while using health information or its proxies for credit decisions may be deemed inappropriate, an exception could be made for proxies that are directly relevant to the credit-worthiness of the individual (e.g., her income and expenses).

1.2.2 Proposed: Proxy Non-discrimination

Analogous to use privacy, proxy non-discrimination constraints restrict the use of protected information types such as gender, race and nationality for purposes such as credit, insurance and healthcare.

Two popular approaches to addressing the problem of discrimination are the prevention of disparate impact and disparate treatment. Disparate impact identifies cases where group parity is violated i.e., where the fraction of individuals who get positive outcomes are very different across protected and unprotected groups in the population. The 80% rule in hiring and promotions is an embodiment of this approach that can be traced back to the Griggs v. Duke Power ruling [15]. However, it has been pointed out that the group parity often does not ensure outcomes which are fair [38]. On the other hand disparate treatment, rules out explicit uses of protected information, which does not rule out inferences of protected information being used. Instead, as with privacy, we take a pragmatic approach of detecting evidence of proxy use of protected information.

Both existing theories allow exceptions to constraints that severely affect the utility of the system. Disparate impact terms these exceptions as *business necessities*, and disparate treatment terms these as *bona fide occupational qualtifications* (BFOQ). In proposed work, described in Section 4.3, we will develop a rigorous language for expressing such exceptions, similar to LEGALEASE.

1.2.3 Proposed: Case Studies in Accountable Information Use

In this proposed task, we will perform two case studies in accountable information use in data-driven system in order to demonstrate the practical viability of the theories and tools for analyzing and repairing proxy use. The first proposed case study will be a predictive policing system, in collaboration with Daniel Neill, who will provide guidance on predictive policing models. He will also provide models developed from the crime and 911 dispatch data from the Pittsburgh PA Bureau of Police, and will evaluate the utility of our mechanisms in this application area. The second case study will use publicly available data for housing mortgages [42] to build an automated loan approval system. Both of these case studies will carefully examine potential use privacy and proxy non-discrimination violations in these systems and attempt to find repairs for violations that don't significantly impact the utility of these systems.

Completed and Proposed Work. In summary, in this report, we present the following completed and ongoing works and propose additional tasks in support of our thesis.

- (Completed) Bootstrapping privacy compliance in big data systems [83] (Chapter 2).
- (Completed) Quantifying explicit use [27] (Chapter 3)
- (Proposed) Distributional faithfulness tradeoffs in causal analysis of machine learning models (Section 3.1, expected completion by Summer 2017)
- (Ongoing) Use privacy in data driven systems (Chapter 4, expected completion by Summer 2017)

- (Proposed) Proxy non-discrimination and expressive use privacy policies (Chapter 4, expected completion by Fall 2017)
- (Proposed) Case studies in accountable information use (Chapter 4, expected completion by Spring 2018).

1.3 Related Work

We now briefly discuss and compare with closely related work. See Chapter 5 for a more comprehensive discussion.

Information Flow Analysis. There has been significant research activity in restricting information flows in programs over the last three decades, and language-based methods to support these restrictions ([30, 68, 75]). These methods enforce non-interference or variants of it from sensitive inputs of the program to outputs. The first portion of our work on bootstrapping privacy compliance in big data systems (Chapter 2), draws ideas from this body of work, and adds a policy specification language that allows specification independent of code, along with a scalable data inventory that allows information flow labels to be bootstrapped without significant human effort.

Work on quantifying information flow has largely focused on quantifying the leakage of information about sensitive attributes to an adversary. Quantitative Information Flow is concerned with information leaks and therefore needs to account for correlations between inputs that may lead to leakage, making measures of associations between inputs and outcomes appropriate (see [85] for an overview). On the other hand, we take the position that information use is a causal notion, and therefore measuring it requires destroying correlations through interventions.

Finally, a line of work on information use and information flow experiments [25, 26, 93], formalizes the relation between causality, probabilistic non-interference, and information use, and develops a framework for black-box experimentation on web services. Black-box experimentation is an important approach to achieving accountability in data driven systems, but is not the focus of this thesis.

Privacy in Statistical Systems. Privacy in the presence of data analytics has largely focused on minimizing the disclosure of personal information. Differential privacy [37] and its variants belong to this class of properties in a setting with a trusted data processor and an untrusted adversary trying to infer sensitive information about individuals. Differential privacy provides the guarantee that any adversary will gain approximately the same information with or without an individual's participation in a dataset. Other formal properties related to privacy focus on limiting the flow of information using notions such as statistical disclosure limitation [41], characterizing possible inferences from data releases [21, 32, 82], or that your participation in a study should not become known [53].

Our notion of use privacy is quite complementary to this body of prior work. Instead of trying to limit disclosures through system outputs, we focus instead on ensuring that protected information types and their proxies are not used internally by the data analytics system, and could be used in conjunction with methods that limit disclosures of sensitive information.

Fairness in Statistical Systems. Recently, the algorithmic foundations of fairness in personal information processing systems have received significant attention [16, 24, 34, 56, 102]. While many of the algorithmic approaches [16, 56, 102] have focused on group parity as a metric for achieving fairness in classification, Dwork et al. [34] argue that group parity is insufficient as a basis for fairness, and propose a similarity-based approach which prescribes that similar individuals should receive similar classification outcomes, along with algorithms for achieving this by design. However, this approach requires a similarity metric for individuals which is often subjective and difficult to construct.

Instead of trying to achieve fairness by design, in our theory of proxy non-discrimination, we attempt to detect and remove instances of discrimination arising out of identifiable explicit or proxy use of protected attributes.

Chapter 2

Qualitative Explicit Use

In this chapter, we briefly describe completed work on the enforcement of explicit use restrictions as a part of an automated privacy compliance workflow in an industrial-scale map reduce system [83]. We interpret restrictions on information use in the sense of *non-interference*, i.e., a data type not supposed to flow to a program should not affect the output of the program. Two important challenges for automating the compliance of such information use properties are that the policies are often written in English by lawyers with limited programming abilities, and the programs are not annotated with the relevant data types. In order to address these challenges our proposed workflow leverages two key technical contributions: (1) a language LEGALEASE for stating privacy policies, which is usable by policy authors, but has precise semantics and enables automated checking for compliance, (2) a self-bootstrapping data inventory mapper GROK, which maps low level data types in code to high-level policy concepts, and bridges the world of product development with the world of policy makers. Specifically, we target privacy compliance of large codebases written in languages that support the Map-Reduce programming model [17, 65, 88]. This focus enables us to apply our approach to current industrial-scale data processing applications, in particular the data analytics backend of Bing, Microsoft's web search engine [1], where it is currently deployed as a part of their privacy compliance process.

Overall, [83] makes three contributions: (i) designs LEGALEASE, an enforceable, expressive, and usable privacy policy language; (ii) designs GROK, a self-bootstrapping, up-to-date, verifiable data inventory for Map-Reduce-like big data systems, (iii) proposes a workflow for automated privacy compliance checking, and (iv) demonstrates that it is feasible to perform and sustain automated privacy compliance checking of existing state-of-the-art big data systems at modest cost.

We now provide an on overview of the two key technical pieces of the workflow: LEGALEASE and GROK, and refer the reader to [83] for more details on the streamlined workflow for privacy compliance checking and its evaluation.

2.1 LEGALEASE

LEGALEASE is a usable, expressive, and enforceable privacy policy language. The primary design criteria for this language were that it (a) be *usable* by the policy authors; (b) be *expressive* enough to capture real privacy policies of industrial-scale systems, e.g., Bing; (c) and should allow *compositional reasoning* on policies.

As the intended users for LEGALEASE are policy authors with limited training in formal languages, enabling usability is essential. To this end, LEGALEASE enforces syntactic restrictions ensuring that encoded policy clauses are structured very similarly to policy texts. Specifically, building on prior work on a first order privacy logic [31], policy clauses in LEGALEASE allow (resp. deny) certain types of information flows and are refined through exceptions that deny (resp. allow) some sub-types of the governed information flow types. This structure of nested allow-deny rules appears in many practical privacy policies, including privacy laws such the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act (GLBA) (as observed in prior work [31]), as well as privacy policies for Bing and Google. A distinctive feature of LEGALEASE (and a point of contrast from prior work based on first-order logic and first order-temporal logic [11, 31]) is that the semantics of policies is compositional: reasoning about a policy is reduced to reasoning about its parts. This form of compositionality is useful because the effect of adding a new clause to a complex policy is locally contained (an exception only refines its immediately enclosing policy clause).

We illustrate LEGALEASE through a series of examples that build up to a complex clause. In the examples we use two user-defined attributes: *DataType* and *UseForPurpose* (our deployment uses two additional ones *AccessByRole* and *InStore*). We define the concept lattice for each of these four attributes in the next subsection.

The simplest LEGALEASE policy is DENY. The policy contains a single clause; the clause contains no exceptions and no attribute restrictions. The policy, rather uninterestingly, simply denies everything. We next add a restriction along the *DataType* attribute for graph nodes to which IP address flows.

```
DENY DataType IPAddress
```

As discussed in our running example, there is often a need to capture some limited form of history of the data flow (e.g., that the IP address has been truncated before it can be used). We capture this notion of typestate in the concept lattice for the *DataType* attribute (described below). The lattice contains an element IPAddress:Truncated meant to represent the truncated IP address, and the lattice element for IP address IPAddress, such that IPAddress:Truncated $\leq$ IPaddress, where $\leq$ is the partial order for the lattice. We next add the exception that allows us to use the truncated IP address. The added lines are marked with ◁.

```
DENY DataType IPAddress
EXCEPT                                                          ◁
    ALLOW DataType IPAddress:Truncated                          ◁
```

The above policy contains a clause with an exception. The first disallows any use of IP address, while the exception relaxes the first allowing use when the IP address is truncated. Next, we restrict the policy to advertising uses only by adding a restriction along the *UseForPurpose* attribute for the value Advertising, while retaining the exception that allows the use of IP Address when truncated.

```
DENY DataType IPAddress
        UseForPurpose Advertising                               ◁
EXCEPT
    ALLOW DataType IPAddress:Truncated
```

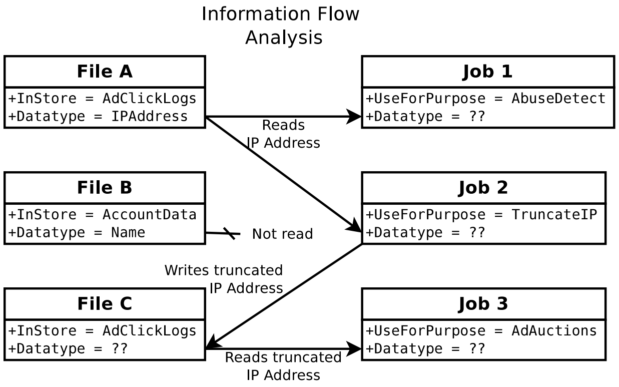


Figure 2.1: Example scenario showing a partially-labeled data dependency graph between three files and programs.

The above policy corresponds to the English clause “full IP address will not be used for advertising”. Note that since the first clause is restricted only to advertising use, and the second rule does not relax that attribute, the net effect is that the clause applies only to use of IP address for advertising and says nothing about non-advertising uses (consistent with the English clause).

Finally, consider the English policy “full IP address will not be used for advertising. IP address may be used for detecting abuse. In such cases it will not be combined with account information.” This policy is encoded in LEGALEASE below. The first, second, and third sentences correspond respectively to lines 1–4, 5–6, and 7–8.

```
DENY DataType IPAddress
        UseForPurpose Advertising
EXCEPT
    ALLOW DataType IPAddress:Truncated
    ALLOW DataType IPAddress                                    ◁
            UseForPurpose AbuseDetect                           ◁
    EXCEPT                                                      ◁
        DENY DataType IPAddress, AccountInfo                    ◁
```

The last clause (in lines 7-8) mentions that the combination of IPAddress and AccountInfo is denied, but these elements can be used individually. It turns out that giving formal semantics to such exceptions where combinations are disallowed whereas individual elements are allowed is non-trivial. We revisit this issue when we give formal semantics to LEGALEASE.

2.2 GROK

The GROK **mapper**. GROK is a data-inventory for Map-Reduce-like big data systems. It maps every dynamic schema-element (e.g., members of a tuple passed between mappers and reducers) to datatypes in LEGALEASE. This inventory can be viewed as a mechanism for annotating existing programs written in languages like Hive [88], Dremel [65], or Scope [17] with the information flow types (datatypes) in LEGALEASE. Our primary design criteria for this inventory were that it (a) be *bootstrapped* with minimal developer effort; (b) reflect *exhaustive and up-to-date* information about all data in the Map-Reduce-like system; and (c) make it easy to *verify* (and update) the mapping from schema-elements to LEGALEASE datatypes.

We use an example of how using GROK, compliance checking is reduced to a form of information flow analysis. Consider the scenario in Fig. 2.1. There are three programs (Jobs 1, 2, 3) and three files (Files A, B, C). Let us assume that the programs are expected to be compliant with a privacy policy considered in the previous section that says: "full IP address will not be used for advertising. IP address may be used for detecting abuse. In such cases it will not be combined with account information." Note that the policy restricts how a certain type of personal information flows through the system. The restriction in this example is based on purpose. Other common restrictions include storage restrictions (e.g., requiring that certain types of user data are not stored together) and, for internal policies, role-based restrictions (e.g., requiring that only specific product team members should use certain types of user data). While our policy language is designed in a general form enabling domain-specific instantiations with different kinds of restrictions, our evaluation of Bing is done with an instantiation that has exactly these three restrictions—purpose, role, and storage—on the use of various types of personal information.

The data dependence graph depicted for the example in Fig. 2.1 provides a useful starting point to conduct the information flow analysis. Nodes in the graph are data stores, processes, and humans. Directed edges represent data flowing from one node to another. To begin, let us assume that programs are labeled with their purpose. For example, Job 1 is for the purpose of AbuseDetect. Furthermore, let us also assume that the source data files are labeled with the type of data they hold. For example, File A holds data of type IPAddress. Given these labels, additional labels can be computed using a simple static dataflow analysis. For example, Job 1 and Job 2 both acquire the datatype label IPAddress since they read File A; File C (and hence Job 3) acquires the datatype label IPAddress:Truncated. Given a labeled data dependence graph, a conservative way of checking non-interference is to check whether there exists a path from restricted data to the program in the data dependence graph. In a programming language such as C or Java, this approach may lead to unmanagable overtainting. Fortunately, the data analytics programs we analyze are written in a restricted programming model without global state and with very limited control flow based on data. Therefore, we follow precisely this approach. Languages like Hive [88], Dremel [65], or Scope [17] that are used to write big data pipelines in enterprises adhere to this programming model. Note, that for the search engine that we analyze, the data dependence graph does not come with these kinds of labels. Bootstrapping these labels without significant human effort is a central challenge addressed by GROK.

Chapter 3

Quantifying Explicit Use

Systems that employ machine learned models in their data analytics pipeline pose a significant challenge to reasoning about how information is used due to their complexity. Many inputs are used as features, and as a result have some causal effect on outcomes, which may be very low. We therefore shift our focus from identifying what information was used to quantifying the degree of use. In [27], we develop a family of measures called Quantitative Input Influence (QII) to quantify the causal influence of inputs of systems on their outcomes. Importantly, these measures provide a foundation for explanations that accompany system decisions (e.g., explaining a specific credit decision) and for testing tools useful for internal and external oversight (e.g., to detect algorithmic discrimination).

Three desiderata drove the definitions of these measures. First, we seek a formalization of a *general* class of transparency reports that allows us to answer many useful transparency queries related to input influence, such as system's decisions about individuals and groups. We achieve this desideratum by formalizing a notion of a *quantity of interest*. QII measures the influence of an input on a quantity of interest. A quantity of interest, denoted by $Q_{\mathcal{A}}(X)$, represents a property of the behavior of the system $\mathcal{A}$, for a given input distribution X. Our formalization supports a wide range of statistical properties including probabilities of various outcomes in the output distribution and probabilities of output distribution outcomes conditioned on input distribution events. Examples of quantities of interest include the conditional probability of an outcome for a particular individual or group, and the ratio of conditional probabilities for an outcome for two different groups (a metric used as evidence of disparate impact under discrimination law in the US [2]).

Second, we seek input influence measures that appropriately account for *correlated inputs*— a common case for our target applications. For example, consider a system that assists in hiring decisions for a moving company. Gender and the ability to lift heavy weights are inputs to the system. They are positively correlated with each other and with the hiring decisions. Yet transparency into whether the system uses the weight lifting ability or the gender in making its decisions (and to what degree) has substantive implications for determining if it is engaging in discrimination (the business necessity defense could apply in the former case [2]). This observation makes us look beyond correlation coefficients and other associative measures. We achieve the second desideratum by formalizing *causal* QII measures. These measures (called *Unary QII*) model the difference in the quantity of interest when the system operates over two related input distributions—the real distribution and a hypothetical (or counterfactual) distribution that is constructed from the real distribution in a specific way to account for correlations among inputs. Specifically, if we are interested in measuring the influence of an input on a quantity of interest of the system behavior, we construct the hypothetical distribution by retaining the original distribution over all other inputs and independently sampling the input of interest from its marginal distribution. The random variable $X_{-i}U_i$ represents the distribution that breaks the correlations between input i and all other inputs and thus lets us measure the influence of this input on the quantity of interest, independently of other correlated inputs. The QII of an input i on a quantity of interest $Q_{\mathcal{A}}$ is defined as:

$$Q_{\mathcal{A}}(X) - Q_{\mathcal{A}}(X_{-i}U_i).$$

Revisiting our moving company hiring example, if the system makes decisions only using the weightlifting ability of applicants, the influence of gender will be zero on the ratio of conditional probabilities of being hired for males and females.

Third, we seek measures that appropriately quantify input influence in settings where any input by itself does not have significant influence on outcomes but a set of inputs does. In such cases, we seek measures of *joint influence* of a set of inputs (e.g., age and income) on a system's decision (e.g., to serve a high-paying job ad). We also seek measures of *marginal influence* of an input within such a set (e.g., age) on the decision. This notion allows us to provide finer-grained transparency about the relative importance of individual inputs within the set (e.g., age vs. income) in the system's decision.

We achieve the third desideratum in two steps. First, we define a notion of joint influence of a set of inputs (called *Set QII*) via a natural generalization of the definition of the hypothetical distribution in the Unary QII definition. Second, we define a family of *Marginal QII* measures that model the difference on the quantity of interest as we consider sets with and without the specific input whose marginal influence we want to measure. Depending on the application, we may pick these sets in different ways, thus motivating several different measures. For example, we could fix a set of inputs and ask about the marginal influence of any given input in that set on the quantity of interest. Alternatively, we may be interested in the average marginal influence of an input when it belongs to one of several different sets that significantly affect the quantity of interest. We consider several marginal influence aggregation measures from cooperative game theory originally developed in the context of influence measurement in voting scenarios and discuss their applicability in our setting. We also build on that literature to present an efficient approximate algorithm for computing these measures.

Recognizing that different forms of transparency reports may be appropriate for different settings, we generalize our QII measures to be parametric in its key elements: the intervention used to construct the hypothetical input distribution; the quantity of interest; the difference measure used to quantify the distance in the quantity of interest when the system operates over the real and hypothetical input distributions; and the aggregation measure used to combine marginal QII measures across different sets. This generalized definition provides a structure for exploring the design space of transparency reports.

Since transparency reports released to an individual, regulatory agency, or the public might compromise individual privacy, we explore the possibility of answering transparency queries while protecting differential privacy [36]. We prove bounds on the sensitivity of a number of transparency queries and leverage prior results on privacy amplification via sampling [57] to accurately answer these queries.

In [27], we demonstrate the utility of the QII framework by developing two machine learning applications on real datasets: an income classification application based on the benchmark `adult` dataset [63], and a predictive policing application based on the National Longitudinal Survey of Youth [3]. Using these applications, we argue the need for causal measurement by empirically demonstrating that in the presence of correlated inputs, observational measures are not informative in identifying input influence. Further, we analyze transparency reports of individuals in our dataset to demonstrate how Marginal QII can provide insights into individuals' classification outcomes. Finally, we demonstrate that under most circumstances, QII measures can be made differentially private with minimal addition of noise, and can be approximated efficiently.

In summary, we makes the following contributions in [27]:

- A formalization of a specific algorithmic transparency problem for decision-making systems. Specifically, we define a family of Quantitative Input Influence metrics that accounts for correlated inputs, and provides answers to a general class of transparency queries, including the absolute and marginal influence of inputs on various behavioral system properties. These metrics can inform the design of transparency mechanisms and guide pro-active system testing and posthoc investigations.
- A formal treatment of privacy-transparency trade-offs, in particular, by construction of differentially private answers to transparency queries.
- An implementation and experimental evaluation of the metrics over two real data sets. The evaluation demonstrates that (a) the QII measures are *informative*; (b) they remain *accurate* while preserving differential privacy; and (c) can be *computed* quite quickly for standard machine learning systems applied to real data sets.

3.1 Proposed Work: Distributional Faithfulness in Causal Analyses of Machine Learning Models

The causal nature of QII and causal testing in general, requires evaluating the system under study on counterfactual inputs. For example, in genetic studies, genes are often artificially removed to study the effect of their absence. Similarly, in QII, we compare the outcomes of the model when one input is changed via an intervention, breaking the correlation with other inputs as a result. However, in a problem peculiar to the causal study of machine learning models, these counterfactual points may lie out of the distribution of inputs the model has been trained to predict on. As a result, the predictions of the model on these counterfactual inputs which lie outside the model's input distribution could be unreliable. In this proposed task we will explore two solutions to this problem. In the first approach, we will define a notion of a faithful counterfactual that lies within the support S of input distribution of the classifier, and define causal measures such that only test against faith counterfactuals. In the notation used in QII, this would restrict us to counterfactuals from the distribution $X_{-i}U_i \mid X_{-i}U_i \in S$. In the second approach, we will explore the effect of training the classifier on counterfactual points. As the model is also trained on these counterfactual points, they are now part of the training distribution, thereby addressing the distributional faithfulness. Since, the labels for the counterfactual points are unknown, we will need an external source to label these counterfactual points, which might be expensive, and therefore we will explore active learning strategies to label these counterfactual points.

For evaluating the efficacy of such a counterfactual active learning algorithm, we will simulate a labeler by choosing a model that represents the ground truth on real world datasets, and train a model to mimic this ground truth.

Task 1. *Address the distributional faithfulness problem in causal analyses of machine learning models using two approaches.*

- *Define faithful causal measures that only compare against counterfactuals belonging to the input distribution of the model.*
- *Design a counterfactual active learning algorithm that trains the model on points from the counterfactual distribution.*

Chapter 4

Proxy Use

While the first part of this work focuses on explicit use, information can be used indirectly through proxies, which allow a data processor to effectively infer protected information types and use them even when they are not explicitly provided.

We use an example, inspired by the Target case [33], to motivate the challenges in defining proxy use in automated decision-making systems. Consider a pharmacy within a retail store, such as Target. We consider various ways in which the retail store and its pharmacy may use information about the pregnancy status, purchases, and credit card type of its customers in making decisions.

The pharmacy *knows* the pregnancy status of its customers (e.g., via a permitted information flow from a doctor's office to the pharmacy with prescription information). However, it is restricted in how it may *use* this information to protect patient privacy. For example, it may legitimately use pregnancy status directly to dispense medicine, but is prohibited from using pregnancy status to target ads. Indeed, this form of use restriction to protect privacy is embodied in privacy laws like the HIPAA Privacy Rule [71] and in many corporate policies (e.g., [48, 66]). They reflect the understanding that knowledge restrictions are inadequate to protect privacy in settings where the knowledge of certain information types may be used to achieve certain desired purposes (e.g., treatment) but not for others (e.g., marketing). Prior work provides methods for enforcing these explicit use restrictions in human and automated decision-making systems (e.g., [83, 93, 95]).

Use restrictions get more nuanced when proxy use comes into play. For example, instead of using the pregnancy status information available to the pharmacy, the retail store could use information in the purchase history that are strong predictors (or proxies) for pregnancy status (e.g., pre-natal vitamins) to target ads. Our goal is to capture this form of proxy use.

In ongoing work [28], we propose a theory of proxy use, and use it as a building block to construct a theory of *use privacy* and *proxy non-discrimination*. In this chapter, we first describe a definition for proxy use in Section 4.1. The development of this definition is guided by axioms that characterize reasonable conditions for proxy use. We then present ongoing and proposed work on how to build theories of use privacy (Section 4.2) and proxy non-discrimination (Section 4.3) using the definition of proxy use as a building block. For more details, we refer the reader to [28], where we describe algorithms for detecting and repairing proxy use, their use in a method for enforcing use privacy, and their evaluation on real datasets.

4.1 Proxy Use

4.1.1 Definitions

We now present an axiomatically justified, formal definition of proxy use in data-driven programs. Our definition for proxy use of a protected information type involves *decomposing* a program to find an intermediate computation whose result exhibits two properties:

- *Proxy*: strong association with the protected type
- *Use*: causal influence on the output of the program

In § 4.1.2, we present a sequence of examples to illustrate the challenge in identifying proxy use in systems that operate on data associated with a protected information type. In doing so, we will also contrast our work with closely-related work in privacy and fairness. In §4.1.3, we formalize the notions of proxy and use, preliminaries to the definition. The definition itself is presented in §4.1.4 and §4.1.5. Finally, in §4.1.6, we provide an axiomatic characterization of the notion of proxy use that guides our definitional choices.

4.1.2 Examples of Proxy Use

Prior work on detecting use of protected information types [24, 45, 60, 91] and leveraging knowledge of detection to eliminate inappropriate uses [45] have treated the system as a black-box. Detection relied either on experimental access to the black-box [24, 60] or observational data about its behavior [45, 91]. Using a series of examples motivated by the Target case, we motivate the need to peer inside the black-box to detect proxy use.

Example 1. *(Explicit use, Fig. 4.1a) A retailer explicitly uses pregnancy status from prescription data available at its pharmacy to market baby products.*

This form of explicit use of a protected information type can be discovered by existing black-box experimentation methods that establish causal effects between inputs and outputs (e.g., see [24, 60]).

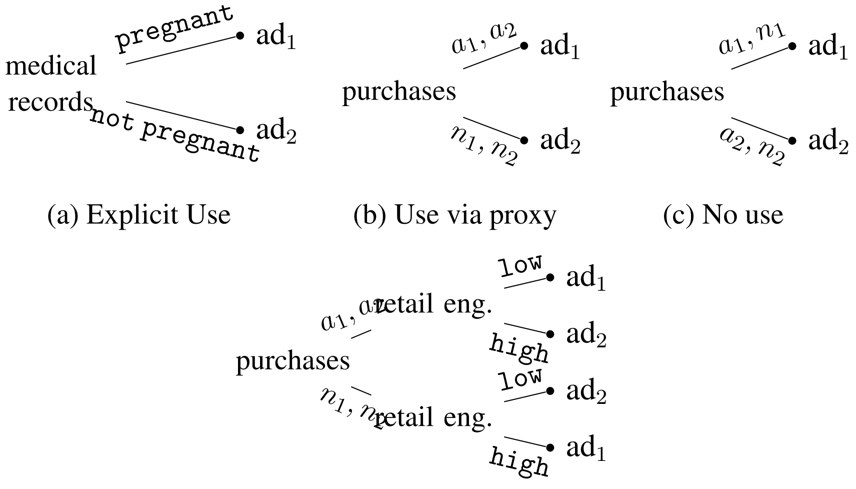
Example 2. *(Inferred use, Fig. 4.1b) Consider a situation where purchase history can be used to accurately predict pregnancy status. A retailer markets specific products to individuals who have recently purchased products indicative of pregnancy (e.g., $a_1, a_2 \in$ purchases).*

This example, while very similar in effect, does not use health information directly. Instead, it infers pregnancy status via associations and then uses it. Existing methods (see [45, 91]) can detect such associations between protected information types and outcomes in observational data.

Example 3. *(No use, Fig. 4.1c) Retailer uses some uncorrelated selection of products ($a_1, n_1 \in$ purchases) to suggest ads.*

In this example, even though the retailer could have inferred pregnancy status from the purchase history, no such inference was used in marketing products. As associations are commonplace, a definition of use disallowing such benign use of associated data would be too restrictive for practical enforcement.

Example 4. *(Masked proxy use, Fig. 4.1d) Consider a more insidious version of Example 2. To mask the association between the outcome and pregnancy status, the company also markets baby*



(a) Explicit Use

(b) Use via proxy

(c) No use

(d) Masked use via proxy

Figure 4.1: Examples of models (decision trees) used by a retailer for offering medicines and for selecting advertisements to show to customers. The retailer uses pregnancy status, past purchases, and customer's level of retail engagement. Products a_1 and a_2 are associated with pregnancy (e.g., prenatal vitamins, scent-free lotions) whereas products n_1 and n_2 are associated with a lack of pregnancy (e.g., alcohol, camping gear); all four products are equally likely. Retail engagement, (`high` or `low`), indicating whether the customer views ads or not, is independent of pregnancy.

products to people who are not pregnant, but have low retail engagement, so these advertisements would not be viewed in any case.

While there is no association between pregnancy and outcome in both Example 3 and Example 4, there is a key difference between them. In Example 4, there is an intermediate computation based on aspects of purchase history that is a predictor for pregnancy status, and this predictor is used to make the decision, and therefore is a case of proxy use. In contrast, in Example 3, the intermediate computation based on purchase history is uncorrelated with pregnancy status. Distinguishing between these examples by measuring associations using black box techniques is non-trivial. Instead, we leverage white-box access to the code of the classifier to identify the intermediate computation that serves as a proxy for pregnancy status. Precisely identifying the particular proxy used also aids the normative decision of whether the proxy use is appropriate in this setting.

4.1.3 Notation and Preliminaries

We assume individuals are drawn from a population distribution $\mathcal{P}$, in which our definitions are parametric. Random variables $W, X, Y, Z, \ldots$ are functions over $\mathcal{P}$, and the notation $W \in \mathcal{W}$ represents that the type of random variable is $W : \mathcal{P} \to \mathcal{W}$. An important random variable used throughout this chapter is $\mathbf{X}$, which represents the vector of features of an individual that is provided to a predictive model. A predictive model is denoted by $\langle \mathbf{X}, \mathcal{A} \rangle_{\mathcal{P}}$, where $\mathcal{A}$ is a

f	A function
$\langle \mathbf{X}, \mathcal{A} \rangle_{\mathcal{P}}$	A model, which is a function $\mathcal{A}$ used for prediction, operating on random variables $\mathbf{X}$, in population $\mathcal{P}$
X	A random variable
p	A program
$\langle \mathbf{X}, p \rangle_{\mathcal{P}}$	A syntactic model, which is a program p, operating on random variables $\mathbf{X}$
$[p_1/X]p_2$	A substitution of p_1 in place of X in p_2
$\mathbf{X}$	A sequence of random variables

Table 4.1: Summary of notation used in the chapter

function that operates on $\mathbf{X}$. For simplicity, we assume that $\mathcal{P}$ is discrete, and that models are deterministic. Table 4.1 summarizes all the notation used in this chapter, in addition to the notation for programs that is introduced later in the chapter.

Proxies

A *proxy* for a random variable Z is a random variable X that is perfectly correlated with Z. Informally, it is possible to use X and Z interchangeably in any function drawing inputs from the same distribution.

Definition 1 (Perfect Proxy)**.** *A random variable $X \in \mathcal{X}$ is a* perfect proxy for $Z \in \mathcal{Z}$ *if there exist functions $f : \mathcal{X} \to \mathcal{Z}, g : \mathcal{Z} \to \mathcal{X}$, such that* $\Pr(Z = f(X)) = \Pr(g(Z) = X) = 1$. □

While this notion of a proxy is too strong in practice, it is useful as a starting point to explain the key ideas in our definition of proxy use. This definition captures two key properties of proxies, *two-sidedness* and *invariance under renaming*.

Two-sidedness Definition 1 captures the property that proxies admit predictors in both directions: it is possible to construct a predictor of X from Z, and vice versa. This strict two-sided association criterion distinguishes benign use of associated information from proxy use as illustrated in the following example.

Example 5. *Recall that in Figure 4.1, a_1, a_2 is a proxy for pregnancy status. In contrast, consider Example 3, where purchase history is an influential input to the program that serves ads to. Suppose that the criteria is to serve ads to those with a_1, n_1 in their purchase history. According to Definition 1, neither purchase history or a_1, n_1 are proxies, because pregnancy status does not predict purchase history or a_1, n_1. However, if Definition 1 were to allow one-sided associations, then purchase history would be a proxy because it can predict pregnancy status. This would have the unfortunate effect of implying that the benign application in Example 3 has proxy use of pregnancy status.* □

Invariance under renaming This definition of a proxy is invariant under renaming of the values of a proxy. Suppose that a random variable evaluates to 1 when the protected information type is 0 and vice versa, then this definition still identifies the random variable as a proxy.

Influence

Our definition of influence aims to capture the presence of a causal dependence between a variable and the output of a function. Intuitively, a variable x is influential on f if it is possible to change the value of f by changing x while keeping the other input variables fixed.

Definition 2. *For a function $f(x, y)$, x is influential if and only if there exists values x_1, x_2, y, such that $f(x_1, y) \neq f(x_2, y)$.* $\square$

In Figure 4.1a, pregnancy status is an influential input of the system, as just changing pregnancy status while keeping all other inputs fixed changes the prediction. Influence, as defined here, is identical to the notion of interference used in the information flow literature.

4.1.4 Definition

We use an abstract framework of program syntax to reason about programs without specifying a particular language to ensure that our definition remains general. Our definition relies on syntax to reason about decompositions of programs into intermediate computations, which can then be identified as instances of proxy use using the concepts described above.

Program decomposition We assume that models are represented by programs. For a set of random variables $\mathbf{X}$, $\langle \mathbf{X}, p \rangle_{\mathcal{P}}$ denotes the assumption that p will run on the variables in $\mathbf{X}$. Programs are given meaning by a denotation function $[\![\cdot]\!]_{\mathbf{X}}$ that maps programs to functions. If $\langle \mathbf{X}, p \rangle_{\mathcal{P}}$, then $[\![p]\!]$ is a function on variables in $\mathbf{X}$, and $[\![p]\!](\mathbf{X})$ represents the random variable of the outcome of p, when evaluated on the input random variables $\mathbf{X}$. Programs support substitution of free variables with other programs, denoted by $[p_1/X]p_2$, such that if p_1 and p_2 programs that run on the variables $\mathbf{X}$ and $\mathbf{X}, X$, respectively, then $[p_1/X]p_2$ is a program that operates on $\mathbf{X}$.

A decomposition of program p is a way of rewriting p as two programs p_1 and p_2 that can be combined via substitution to yield the original program.

Definition 3 (Decomposition)**.** *Given a program p, a decomposition (p_1, X, p_2) consists of two programs p_1, p_2, and a fresh variable X, such that $p = [p_1/X]p_2$.* $\square$

For the purposes of our proxy use definition we view the first component p_1 as the intermediate computation suspected of proxy use, and p_2 as the rest of the computation that takes in p_1 as an input.

Definition 4 (Influential Decomposition)**.** *Given a program p, a decomposition (p_1, X, p_2) is influential iff X is influential in p_2.* $\square$

Main definition

Definition 5 (Proxy Use)**.** *A program $\langle \mathbf{X}, p \rangle_{\mathcal{P}}$ has proxy use for a random variable Z if there exists an influential decomposition (p_1, X, p_2) of $\langle \mathbf{X}, p \rangle_{\mathcal{P}}$, and $[\![p_1]\!](\mathbf{X})$ is a proxy for Z.* $\square$

Example 6. *In Figure 4.1d, this definition would identify proxy use using the decomposition (p_1, U, p_2), where p_2 is the entire tree, but with the condition $(a_1, a_2 \in \text{purchases})$ replaced by* *the variable U. In this example, U is influential in p_2, since changing the value of U changes the outcome. Also, we assumed that the condition $(a_1, a_2 \in \mathit{purchases})$ is a perfect predictor for pregnancy, and is therefore a proxy for pregnancy. Therefore, according to our definition of proxy use, the model in 4.1d has proxy use of pregnancy status.*

4.1.5 A Quantitative Relaxation

Definition 5 is too strong in one sense and too weak in another. It requires that intermediate computations be perfectly correlated with a protected attribute, and that there exists *some* input, however improbable, in which the result of the intermediate computation is relevant to the model. For practical purposes, we would like to capture imperfect proxies that are strongly associated with an attribute, but only those whose influence on the final model is appreciable. To relax the requirement of perfect proxies and non-zero influence, we quantify these two notions to provide a parameterized definition.

ϵ-proxies We wish to measure how strongly a random variable X is a proxy for a random variable Z. Recall the two key requirements from the earlier definition of a proxy: (i) the association needs to be two-sided, and (ii) the association needs to be invariant under renaming of the random variables. The *variation of information metric* $d_{\text{var}}(X, Z) = H(X|Z) + H(Z|X)$ [20] is one measure that satisfies these two requirements. The first component in the metric, the conditional entropy of X given Z, $H(X|Z)$, measures how well X can be predicted from Z, and $H(Z|X)$ measures how well Z can be predicted from X, thus satisfying the requirement for the metric being two-sided. Additionally, one can show that conditional entropies are invariant under renaming, thus satisfying our second criteria. To obtain a normalized measure in $[0, 1]$, we choose $1 - \frac{d_{\text{var}}(X,Z)}{H(X,Z)}$ as our measure of association, where the measure being 1 implies perfect proxies, and 0 implies statistical independence. Interestingly, this measure is identical to normalized mutual information [20], a standard measure that has also been used in prior work in identifying associations in outcomes of machine learning models [91].

Definition 6 (Proxy Association)**.** *Given two random variables X and Z, the strength of a proxy is given by normalized mutual information,*

$$d(X, Z) = 1 - \frac{H(X|Z) + H(Z|X)}{H(X, Z)}$$

where X is defined to be an ϵ-proxy for Z if $d(X, Z) \geq \epsilon$.

δ-influential decomposition Recall that for a decomposition (p_1, X, p_2), in the qualitative sense, influence is given by interference which implies that there exists x, x_1, x_2, such that $[\![p_2]\!](x_1, x) \neq [\![p_2]\!](x_2, x)$. Here x_1, x_2 are values for the output of p_1, that for a given x, change the outcome of p_2. However, this definition is too strong as it requires only a single pair of values x_1, x_2 to show that the outcome can be changed by p_1 alone. To measure influence, we quantify interference by using Quantitative Input Influence (QII), a causal measure of input influence introduced in [27]. In our context, for a decomposition (p_1, X, p_2), the influence of p_1 on p_2 is given by:

$$\iota(p_1, p_2) = \mathbb{E}_{\mathbf{X},\mathbf{X}' \xleftarrow{\$} \mathcal{P}}(\llbracket p \rrbracket(\mathbf{X}) \neq \llbracket p_2 \rrbracket(\mathbf{X}, \llbracket p_1 \rrbracket(\mathbf{X}'))).$$

Intuitively, this quantity measures the likelihood of finding randomly chosen values of the output of p_1 that would change the outcome of p_2.

Definition 7 (Decomposition Influence)**.** *Given a decomposition (p_1, X, p_2), the influence of the decomposition is given by the QII of X on p_2. A decomposition (p_1, X, p_2) is defined to be δ-influential if $\iota(p_1, p_2) > \delta$.*

(ϵ, δ)**-proxy use** Now that we have quantitative versions of the primitives used in Definition 5, we are in a position to define quantitative proxy use (Definition 8). The structure of this definition is the same as before, with quantitative measures substituted in for the qualitative assertions used in Definition 5.

Definition 8 ((ϵ, δ)-proxy use)**.** *A program $\langle \mathbf{X}, p \rangle_{\mathcal{P}}$ has (ϵ, δ)-proxy use of random variable Z iff there exists a δ-influential decomposition (p_1, X, p_2), such that $\llbracket p \rrbracket(\mathbf{X})$ is an ϵ-proxy for Z.*

This definition is a strict relaxation of Definition 5, which reduces to $(1, 0)$-proxy use.

4.1.6 Axiomatic Basis for Definition

We now motivate our definitional choices by reasoning about a natural set of properties that a notion of proxy use should satisfy. We first prove an important impossibility result that shows that no definition of proxy use can satisfy four natural semantic properties of proxy use. The central reason behind the impossibility result is that under a purely semantic notion of function composition, the causal effect of a proxy can be made to disappear. Therefore, we choose a syntactic notion of function composition for the definition of proxy use presented above. The syntactic definition of proxy use is characterized by syntactic properties which map very closely to the semantic properties.

Property 1. (Explicit Use) *If Z is an influential input of the model $\langle \{\mathbf{X}, Z\}, \mathcal{A} \rangle_{\mathcal{P}}$, then $\langle \{\mathbf{X}, Z\}, \mathcal{A} \rangle_{\mathcal{P}}$ has proxy use of Z.*

This property identifies the simplest case of proxy use: if an input to the model is influential, then the model exhibits proxy use of that input.

Property 2. (Preprocessing) *If a model $\langle \{\mathbf{X}, X\}, \mathcal{A} \rangle_{\mathcal{P}}$ has proxy use of random variable Z, then for any function f such that $\Pr(f(\mathbf{X}) = X) = 1$, let $\mathcal{A}'(x) = \mathcal{A}(x, f(x))$. Then, $\langle \mathbf{X}, \mathcal{A}' \rangle_{\mathcal{P}}$ has proxy use of Z.*

This property covers the essence of proxy use where instead of being provided a protected information type explicitly, the program uses a strong predictor for it instead. This property states that models that use inputs explicitly and via proxies should not be differentiated under a reasonable theory of proxy use.

Property 3. (Dummy) *Given $\langle \mathbf{X}, \mathcal{A} \rangle_{\mathcal{P}}$, define $\mathcal{A}'$ such that for all x, x', $\mathcal{A}'(x, x') = \mathcal{A}(x)$, then $\langle \mathbf{X}, \mathcal{A} \rangle_{\mathcal{P}}$ has proxy use for some Z iff $\langle \{\mathbf{X}, X\}, \mathcal{A}' \rangle_{\mathcal{P}}$ has proxy use of Z.*

This property states that the addition of an input to a model that is not influential, i.e., has no effect on the outcomes of the model, has no bearing on whether a program has proxy use or not. This property is an important sanity check that ensures that models aren't implicated by the inclusion of inputs that they do not use.

Property 4. (Independence) *If $\mathbf{X}$ is independent of Z in $\mathcal{P}$, then $\langle \mathbf{X}, \mathcal{A}\rangle_{\mathcal{P}}$ does not have proxy use of Z.*

Independence between the protected information type and the inputs ensures that the model cannot infer the protected information type for the population $\mathcal{P}$. This property captures the intuition that if the model cannot infer the protected information type then it cannot possibly use it.

While all of these properties seem intuitively desirable, it turns out that these properties can not be achieved simultaneously.

Theorem 1. *No definition of proxy use can satisfy Properties 1-4 simultaneously.*

Proof. Proof by contradiction. Assume that there exists a definition of proxy usage that satisfies all four properties. Let $\mathcal{X} = \{0, 1\}$, and X is a uniform Bernoulli variable over $\mathcal{X}$. The model $\mathcal{A}(x) = x$ is the identity function. Let Z be an independent uniform Bernoulli variable. According to (independence), $\mathcal{A}$ has no proxy usage of Z. Choose $\mathcal{A}'(x, z) = \mathcal{A}(x)$ which operates over $\mathcal{X} \times \mathcal{Z}$. By (dummy), $\mathcal{A}'$ has no implicit use of Z. We choose the following bijective transformation: $f(x, z) = (u, z) = (x \oplus z, z)$, and $f^{-1}(u, z) = (u \oplus z, z)$ In this transformed space, we choose $\mathcal{A}'' = \mathcal{A}' \circ f^{-1}$. Therefore, $\mathcal{A}''(u, z) = u \oplus z$, since $\mathcal{A}''(u, z) = \mathcal{A}'(f^{-1}(u, z)) = \mathcal{A}'(u \oplus z, z) = u \oplus z$. According to (representation independence), $\mathcal{A}''$ has no implicit use of Z. However, since z is an influential input of the model, according to (explicit use of proxy), $\mathcal{A}''$ has implicit use of Z. Therefore, we have a contradiction. □

The key intuition behind this result is that Property 2 requires proxy use to be preserved when an input is replaced with a function that predicts that input via composition. However, with a purely semantic notion of function composition, after replacement, the proxy may get canceled out. To overcome this impossibility result, we choose a more syntactic notion of function composition, which is tied to how the function is represented as a program, and looks for evidence of proxy use within the representation.

We now proceed to the axiomatic justification of our definition of proxy use. As in our attempt to formalize a semantic definition, we base our definition on a set of natural properties given below. These are syntactic versions of their semantic counterparts defined earlier.

Property 5. (Syntactic Explicit Use) *If X is a proxy of Z, and X is an influential input of $\langle \{\mathbf{X}, X\}, p\rangle_{\mathcal{P}}$, then $\langle \{\mathbf{X}, X\}, p\rangle_{\mathcal{P}}$ has proxy use.*

Property 6. (Syntactic Preprocessing) *If $\langle \{\mathbf{X}, X\}, p_1\rangle_{\mathcal{P}}$ has proxy use of Z, then for any p_2 such that* $\Pr\left([\![p_2]\!](\mathbf{X}) = X\right) = 1$, *$\langle \mathbf{X}, [p_2/X]p_1\rangle_{\mathcal{P}}$ has proxy use of Z.*

Property 7. (Syntactic Dummy) *Given a program $\langle \mathbf{X}, p\rangle_{\mathcal{P}}$, $\langle \mathbf{X}, p\rangle_{\mathcal{P}}$ has proxy use for some Z iff $\langle \{\mathbf{X}, X\}, p\rangle_{\mathcal{P}}$ has proxy use of Z.*

Property 8. (Syntactic Independence) *If $\mathbf{X}$ is independent of Z, then $\langle \mathbf{X}, p\rangle_{\mathcal{P}}$ does not have proxy use of Z.*

Properties 5 and 6 together characterize a complete inductive definition, where the induction is over the structure of the program. Suppose we can decompose programs p into (p_1, X, p_2) such that $p = [p_1/X]p_2$. Now if X, which is the output of p_1, is a proxy for Z and is influential in p_2, then by Property 5, p_2 has proxy use. Further, since $p = [p_1/X]p_2$, by Property 6, p has proxy use. This inductive definition where we use Property 5 as the base case and Property 6 for the induction step, precisely characterizes Definition 5. Additionally, it is can be shown that

Definition 5 also satisfies Properties 7 and 8. Essentially, by relaxing our notion of function composition to a syntactic one, we obtain a practical definition of proxy use characterized by the natural axioms above.

4.2 Use Privacy

We return to the Target example described earlier in the chapter to motivate our notion of use privacy. Historically, data collected in a context of interaction between a retailer and a consumer is not expected to result in flows of health information. However, such flow constraints considered in significant theories of privacy (e.g., see Nissenbaum [69]) cannot be enforced because of possible statistical inferences. In particular, prohibited information types (e.g., pregnancy status) could be inferred from legitimate flows (e.g., shopping history). Thus, the theory of use privacy instead ensures that the data processing systems "simulate ignorance" of protected information types (e.g., pregnancy status) and their proxies (e.g., purchase history) by not using them in their decision-making. Because not all instances of proxy use of a protected information type are inappropriate, our theory of use privacy makes use of a normative judgment oracle that makes this inappropriateness determination for a given instance.

We model the personal data processing system as a program p. The use privacy constraint governs a protected information type Z. Our definition of use privacy makes use of two building blocks: (1) a function that given p, Z, and a population distribution $\mathcal{P}$ returns a witness w of proxy use of Z in a program p (if it exists); and (2) a normative judgment oracle $\mathcal{O}(w)$ that given a specific witness returns a judgment on whether the specific proxy use is appropriate (TRUE) or not (FALSE).

Definition 9 (Use Privacy)**.** *Given a program p, protected information type Z, normative judgment oracle $\mathcal{O}$, and population distribution $\mathcal{P}$, use privacy in a program p is violated if there exists a witness w in p of proxy use of Z in $\mathcal{P}$ such that $\mathcal{O}(w)$ returns* FALSE.

In this work, we formalize the computational component of the above definition of use privacy, by using our definition of proxy use which formalizes what it means to use an information type directly or through proxies and design an algorithm to detect proxy uses in programs. We assume that the normative judgment oracle is given to us and use it to identify inappropriate proxy uses and then repair them.

This definition cleanly separates computational considerations that are automatically enforceable and ethical judgments that require input from human experts. This form of separation exists also in some prior work on privacy [47] and fairness [34].

Task 2 (Ongoing)**.** *Develop a theory of use privacy, along with mechanisms for detection and repair built on the definition of proxy use.*

4.3 Proxy Non-discrimination

The theory of proxy non-discrimination prohibits the proxy use of membership in a protected class for certain decisions. Currently, such restrictions for protected classes based on gender or race are required by the law for education, credit, and employment. Indeed, our treatment of proxy use combines elements of causation and association found in two different parts of anti-discrimination law in the US adapted to the setting of automated decision making systems.

Title VII of U.S. Civil Rights Act prohibits use of race, sex, and other protected attributes for employment decisions [4]. Similar laws govern credit [43] and housing decisions [58]. The case law on enforcing these laws has developed various definitions of when such a protected attribute is *used* for a decision.

Direct *disparate treatment*, on the one hand, corresponds to the obvious case: purposefully and directly using the value of a person's race or sex as an input to a decision-making process, a causal property of the process given that data is unlikely to be provided accidentally to such a process. *Disparate impact*, on the other hand, occurs when the same rule is applied to the protected class without regard for class membership but results in significantly worse outcomes for that class. The courts and regulators have used a variety of heuristics and statistical methods to define "significantly". The most well known is 80% rule, which requires that the rate of hiring of a protected class should be within 80% of the rest [99]. These significance tests each measure the degree of association, but not necessarily causation, between membership in a protected class and employment outcomes.

The courts also recognize more subtle indirect usage, such as pretextually using neighborhood (redlining) or education level as a proxy for race [76]. Analogous to use privacy, our definition of proxy use allows for the formalization of such indirect uses. A further complication is that employers can defend themselves against disparate impact by showing that the difference arose due to a *business necessity*. For example, a moving company may require employees be able to lift 200lbs, a requirement yielding a disparate impact on women, but possibly justifiable as a business necessity. Thus, for automated testing of disparate impact to scale, it will require some method of screening out suspected cases that are justified by a business necessity. Similarly, the theory of disparate treatment contains exceptions for *bona fide occupational qualifications* for gender.

> Formalize proxy non-discrimination as restrictions on proxy use of membership in a protected class, with justified exceptions guided by utility considerations.

Consider a case of external auditors discovering associations between race and outcomes in a system that predicts the risk of recidivism used in sentencing systems (as in Angwin et al. [7]). Using a theory of proxy discrimination will allow an analyst to identify proxies that explain the presence of these associations, and then make fine-grained judgements of whether the use of these proxies is justified for predicting recidivism.

While the theory of proxy non-discrimination and use privacy are mathematically isomorphic, this proposed task includes two additional pieces of work. First, the theory of proxy use presented above only refers to proxy use at the level of a population, whereas individual harms arising out of improper use are not accounted for. For instance, in the target case, the individuals for whom the proxy was influential were harmed by the use. Further, to express complex domain specific exceptions in policies, we will develop an extension to LEGALEASE (presented in Chapter 2) with semantics that also support proxy use.

Task 3. *Develop a theory of proxy non-discrimination, built on the definition of proxy use with the following features:*

- *A treatment of proxy use, centered around harms to an individual.*

- *A language to express information use policies that also pertain to proxy use.*

4.4 Case Studies in Accountable Information Use

In this proposed task, we will perform two case studies in accountable information use in data-driven system in order to demonstrate the practical viability of the theories and tools for analyzing and repairing proxy use. The first proposed case study will be a predictive policing system, in collaboration with Daniel Neill, who will provide guidance on predictive policing models. He will also provide models developed from the crime and 911 dispatch data from the Pittsburgh PA Bureau of Police, and will evaluate the utility of our mechanisms in this application area. The second case study will use publicly available data for housing mortgages to build an automated loan approval system [42]. Both of these case studies will carefully examine potential use privacy and proxy non-discrimination violations in these systems and attempt to find repairs for violations that don't significantly impact the utility of these systems.

Task 4. *Perform two case studies in accountable information use for a predictive policing system, and an automated loan approval system.*

Chapter 5

Related Work

In this section we compare with related work on analyzing explicit information use (Section 5.1), and information use via proxies along with related theories of privacy and fairness(Section 5.2). Additionally, we compare with an emerging body of work on explaining machine learning systems (Section 5.3).

5.1 Explicit Use

5.1.1 Qualitative Explicit Use

Information flow analysis of programs There has been significant work in restricting information flows in programs over the last three decades [30] and on language-based methods that support these restrictions, including languages like Jif [68], which augments Java with information flow types, and Flow Caml, which augments ML [75] (see [80] for a survey of these and other language-based methods). These languages can enforce information flow properties like non-interference with mostly static type checking. Taking Jif as one example language, we note that prior work has shown that Jif principals can be used to model role-based [68] and purpose-based [52] restrictions on information flow. Additionally, recognizing that non-interference is too strong a requirement, the theory of relaxed non-interference through declassification [19, 62, 81], allows expressing policies that, for instance, do not allow disclosure of individual ages, but allow the disclosure of average age. This line of work also includes techniques for automated inference of declassification policies [92, 101] with minimal programmer annotations. While these ideas have parallels in our work, there are also some significant differences. First, our policy language LEGALEASE enables explicit specification of policies separately from the code whereas in language-based approaches like Jif the policies are either expressed implicitly via typed interface specifications or explicitly via conditionals on program variables. The separation of high-level policy specification from code is crucial in our setting since we want the first task to be accessible to privacy champions and lawyers. Second, since our goal is to bootstrap compliance checking on existing code, we do not assume that the code is annotated with information flow labels. A central challenge (addressed by GROK) is to bootstrap these labels without significant human effort. Once the labels are in place, information flow analysis for our restricted programming model is much simpler than it is for more complex languages like Jif. Note that we (as well as Hayati and Abadi [52]) assume that programs are correctly annotated with their purposes. A semantic definition of what it means for an agent (a program or human) to use information for a purpose is an orthogonal challenge, addressed in part in other work [94].

Privacy policy enforcement over executions A second line of work checks executions of systems (i.e., traces of actions produced by programs or humans) for compliance with privacy policies that restrict how personal information may flow or be used. This line of work includes auditing, run-time monitoring, and logic programming methods for expressive fragments of first-order logic and first-order temporal logics [11, 12, 13, 46] applied to practical policies from healthcare, finance and other sectors. These results are different from ours in two ways. First, their language of restrictions on information flow is more expressive than ours—they can encode role-based and purpose-based restrictions much like we do, but can express a much larger class of temporal restrictions than we can in LEGALEASE with our limited typestates on data. Second, since their enforcement engines only have access to executions and not the code of programs, they can only check for direct flows of information and not non-interference-like properties. Such code analysis is also a point of difference from enforcement using reference monitors of access control and privacy policy languages—an area in which there is a large body of work, including languages such as XACML [67] and EPAL [8].

5.1.2 Quantifying Information Use

Quantitative Information Flow One can think of our results as a causal alternative to *quantitative information flow*. Quantitative information flow is a broad class of metrics that quantify the information leaked by a process by comparing the *information* contained before and after observing the outcome of the process. Quantitative Information Flow traces its information-theoretic roots to the work of Shannon [84] and Rényi [77]. Recent works have proposed measures for quantifying the security of information by measuring the amount of information leaked from inputs to outputs by certain variables; we point the reader to [85] for an overview, and to [20] for an exposition on information theory. Quantitative Information Flow is concerned with information leaks and therefore needs to account for correlations between inputs that may lead to leakage. The dual problem of transparency, on the other hand, requires us to destroy correlations while analyzing the outcomes of a system to identify the causal paths for information leakage.

Experimentation on Web Services There is an emerging body of work on systematic experimentation to enhance transparency into Web services such as targeted advertising [9, 26, 49, 59, 60]. The setting in this line of work is different since they have restricted access to the analytics systems through publicly available interfaces. As a result they only have partial control of inputs, partial observability of outputs, and little or no knowledge of input distributions. The intended use of these experiments is to enable external oversight into Web services without any cooperation. Our framework is more appropriate for a transparency mechanism where an entity proactively publishes transparency reports for individuals and groups. Our framework is also appropriate for use as an internal or external oversight tool with access to mechanisms with control and knowledge of input distributions, thereby forming a basis for testing.

Quantitative Causal Measures Causal models and probabilistic interventions have been used in a few other settings. While the form of the interventions in some of these settings may be very similar, our generalization to account for different quantities of interests enables us to reason about a large class of transparency queries for data analytics systems ranging from classification outcomes of individuals to disparity among groups. Further, the notion of marginal contribution which we use to compute responsibility does not appear in this line of prior work.

Janzing et al. [55] use interventions to assess the causal importance of relations between variables in causal graphs; in order to assess the causal effect of a relation between two variables, $X \rightarrow Y$ (assuming that both take on specific values $X = x$ and $Y = y$), a new causal model is constructed, where the value of X is replaced with a prior over the possible values of X. The influence of the causal relation is defined as the KL-Divergence of the joint distribution of all the variables in the two causal models with and without the value of X replaced. The approach of the intervening with a random value from the prior is similar to our approach of constructing X_{-S}.

Independently, there has been considerable work in the machine learning community to define importance metrics for variables, mainly for the purpose of feature selection (see [50] for a comprehensive overview). One important metric is called Permutation Importance [14], which measures the importance of a feature towards classification by randomly permuting the values of the feature and then computing the difference of classification accuracies before and after the permutation. Replacing a feature with a random permutation can be viewed as a sampling the feature independently from the prior.

There exists extensive literature on establishing causal relations, as opposed to quantifying them. Prominently, Pearl's work [73] provides a mathematical foundation for causal reasoning and inference. In [89], Tian and Pearl discuss measures of causal strength for individual binary inputs and outputs in a probabilistic setting. Another thread of work by Halpern and Pearl discusses actual causation [51], which is extended in [18] to derive a measure of responsibility as degree of causality. In [18], Chockler and Halpern define the responsibility of a variable X to an outcome as the amount of change required in order to make X the counterfactual cause.

5.2 Proxy Use, and Privacy and Fairness for Data-Driven Systems

Our theories of privacy and fairness: use privacy and proxy non-discrimination, are built on our definition of proxy use, which requires the existence of an internal computation that is strongly associated with a protected information type and has causal influence on the outcome. Much of the prior work either aims to minimize (i) associations of outcomes with the protected information type, or (ii) the causal influence of protected inputs on outcomes. We discuss in Chapter 4, how, in many situations these criteria are not necessary or sufficient for identifying improper use of information. Additionally, precisely identifying internal computations that are evidence of proxy use, we can make fine-grained normative judgments about them and repair them if necessary. However, because of the focus on internal computations, our definitions apply to a setting where an auditor has access to programs. We describe how these distinctions apply to specific theories of privacy and fairness, and then discuss algorithmic considerations in the detection and repair of proxy use.

5.2.1 Privacy

Minimizing disclosures Privacy in the presence of data analytics has largely focused on minimizing the disclosure of personal information. Differential privacy [37] and its variants belong to this class of properties in a setting with a trusted data processor and an untrusted adversary trying to infer sensitive information about individuals. Differential privacy provides the guarantee that any adversary will gain approximately the same information with or without an individual's participation in a dataset. Other formal properties related to privacy focus on limiting the flow of information using notions such as statistical disclosure limitation [41], characterizing possible inferences from data releases [21, 32, 82], or that your participation in a study should not become known [53].

Our notion of use privacy is quite complementary to this body of prior work. Instead of trying to limit disclosures through system outputs, we focus instead on ensuring that protected information types and their proxies are not used internally by the data analytics system. Indeed, in many settings it may be desirable to provide both use privacy and disclosure privacy for different sets of principals. For example, when machine learning models are trained using personal data, it is desirable to minimize disclosures pertaining to individuals in the training set, and reducing the use of protected information types for the individuals the models are applied to.

Identifying explicit use The privacy literature on use restrictions has typically focused on explicit use of protected information types, not on proxy use (see Tschantz et al. [93] for a survey and Lipton and Regan [64]). Recent work on discovering personal data use by black-box web services focuses mostly on explicit use of protected information types by examining causal effects [25, 60]; some of this work also examines associational effects [59, 60]. Associational effects capture some forms of proxy use but not others as we argued in Section 4.1.

5.2.2 Fairness

The algorithmic foundations of fairness in personal information processing systems have received significant attention recently [16, 24, 34, 56, 74, 102]. While many of the algorithmic approaches [16, 56, 102] have focused on group parity as a metric for achieving fairness in classification, Dwork et al. [34] argue that group parity is insufficient as a basis for fairness, and propose a similarity-based approach which prescribes that similar individuals should receive similar classification outcomes. However, this approach requires a similarity metric for individuals which is often subjective and difficult to construct.

Proxy Influence Adler et al. [6] quantify the indirect influence of an attribute by obscuring the attribute (along with associations) from a dataset and comparing the prediction accuracy of a model before and after obscuring. This approach does not distinguish between allowed and prohibited proxy use, and therefore is not able to form a basis for normative judgements on permitted proxy use, as required for determining privacy and fairness for systems.

5.2.3 Detection and Repair Models

Our definition of proxy use operates with white-box access to the prediction model. Prior work requires weaker access assumptions.

Access to observational data Detection techniques working under an associative use definition [45, 91] usually only require access to observational data about the behavior of the system.

Access to black-box experimental data Detection techniques working under an explicit use definition of information use [25, 60] typically require experimental access to the system. This access allows the analyst to control some inputs to the system and observe relevant outcomes.

The stronger access level allows us to decompose the model and trace an intermediate computation that is a proxy. Such traceability is not afforded by the weaker access assumptions in prior work. Thus, we explore a different point in the space by giving up on the weaker access requirement to gain the ability to trace and repair proxy use.

Tramèr et al. [91] solve an important orthogonal problem of efficiently identifying populations where associations may appear. Since our definition is parametric in the choice of the population, their technique could allow identifying useful populations to apply our methods to.

5.3 Explaining Machine Learning Systems

Interpretable Machine Learning An orthogonal approach to adding interpretability to machine learning is to constrain the choice of models to those that are interpretable by design. This can either proceed through regularization techniques such as Lasso [90] that attempt to pick a small subset of the most important features, or by using models that structurally match human reasoning such as Bayesian Rule Lists [61], Supersparse Linear Integer Models [100], or Probabilistic Scaling [79]. Since the choice of models in this approach is restricted, a loss in predictive accuracy is a concern, and therefore, the central focus in this line of work is the minimization of the loss in accuracy while maintaining interpretability. On the other hand, our approach to interpretability is forensic. We add interpretability to machine learning models after they have been learnt. As a result, our approach does not constrain the choice of models that can be used.

Simplified Retraining In [78], Ribeiro et al., introduce an approach called Locally Interpretable Model-Agnostic Explanations (LIME), which first maps the input space to an interpretable space, and learns a simple model in the interpretable space that only trained in a small neighborhood around an instance in the interpretable space. In the case where the simple model is a linear model, the coeffients of the model serve as an explanation for the classification for the instance. While the mapping to the interpretable space allows them to generate explanations for a larger set of models such as deep networks. However, the use of machine learning in generating the explanation means that the causal connection between the model and the explanation is unclear. Additionally, our axiomatic approach to defining QII provides theoretical justification for the use of Shapley value for QII.

Game-Theoretic Influence Measures Recent years have seen game-theoretic influence measures used in various settings. Datta et al. [23] also define a measure for quantifying feature influence in classification tasks. Their measure does not account for the prior on the data, nor does it use interventions that break correlations between sets of features. In our terminology, the quantity of interest used by [23] is the ability of changing the outcome by changing the state of a feature. This work greatly extends and generalizes the concepts presented in [23], by both accounting for interventions on sets, and by generalizing the notion of influence to include a wide range of system behaviors, such as group disparity, group outcomes and individual outcomes.

Bibliography

[1] Bing. URL `http://www.bing.org/`. 1.1.1, 2

[2] E.G. Griggs v. Duke Power Co., 401 U.S. 424, 91 S. Ct. 849, 28 L. Ed. 2d 158 (1977). 3

[3] National longitudinal surveys. `http://www.bls.gov/nls/`. 3

[4] Title vii of the civil rights act of 1964, 1964. URL `https://www.eeoc.gov/laws/statutes/titlevii.cfm`. Accessed Aug 13, 2016. 1, 4.3

[5] Equal Credit Opportunity Act (ECOA), 1974. URL `https://www.justice.gov/crt/equal-credit-opportunity-act-3`. Accessed Feb 24, 2017. 1

[6] Philip Adler, Casey Falk, Sorelle Friedler, Gabriel Rybeck, Carlos Schedegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. In *Proceedings of the 2016 IEEE International Conference on Data Mining (ICDM)*, ICDM '16, pages 339–348, Washington, DC, USA, 2016. IEEE Computer Society. ISBN 978-1-4673-9504-5. 5.2.2

[7] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and itâĂŹs biased against blacks. *ProPublica*, May 2016. URL `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`. 1, 4.3

[8] Paul Ashley, Satoshi Hada, Günter Karjoth, Calvin Powers, and Matthias Schunter. Enterprise privacy authorization language (epal 1.2). *Submission to W3C*, 2003. 5.1.1

[9] Paul Barford, Igor Canadi, Darja Krushevskaja, Qiang Ma, and S. Muthukrishnan. Adscape: Harvesting and analyzing online display ads. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 597–608, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. 5.1.2

[10] S. Barocas and H. Nissenbaum. Big data's end run around procedural privacy protections. *Communications of the ACM*, 57(11):31–33, October 2014. 1

[11] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: framework and applications. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15 pp.–198, 2006. doi: 10.1109/SP.2006.32. 2.1, 5.1.1

[12] David A. Basin, Felix Klaedtke, Samuel Müller, and Birgit Pfitzmann. Runtime monitoring of metric first-order temporal properties. In *FSTTCS*, pages 49–60, 2008. 5.1.1

[13] David A. Basin, Felix Klaedtke, Srdjan Marinovic, and Eugen Zalinescu. Monitoring compliance policies over incomplete and disagreeing logs. In *RV*, pages 151–167, 2012.

5.1.1

[14] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125. 5.1.2

[15] Warren Burger. Griggs v. duke power company. Opinion of the United States Supreme Court, March 1971. 1.2.2

[16] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010. ISSN 1384-5810. 1.3, 5.2.2

[17] Ronnie Chaiken, Bob Jenkins, Per-Ake Larson, Bill Ramsey, Darren Shakib, Simon Weaver, and Jingren Zhou. Scope: easy and efficient parallel processing of massive data sets. *Proc. VLDB Endow.*, 1(2):1265–1276, August 2008. ISSN 2150-8097. 1.1.1, 2, 2.2

[18] H. Chockler and J.Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004. 5.1.2

[19] Stephen Chong and Andrew C. Myers. Security policies for downgrading. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS ’04, pages 198–209, New York, NY, USA, 2004. ACM. ISBN 1-58113-961-6. doi: 10.1145/1030083.1030110. 5.1.1

[20] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 4.1.5, 5.1.2

[21] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977. 1.3, 5.2.1

[22] Data Protection Commissioner, Ireland. Facebook ireland ltd, report of re-audit, 2012. URL `http://www.dataprotection.ie/documents/press/Facebook_Ireland_Audit_Review_Report_21_Sept_2012.pdf`. 1.1.1

[23] A. Datta, A. Datta, A.D. Procaccia, and Y. Zick. Influence in classification via cooperative game theory. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 511–517, 2015. 5.3

[24] A. Datta, M.C. Tschantz, and A. Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proceedings on Privacy Enhancing Technologies (PoPETs 2015)*, pages 92âĂŞ–112, 2015. 1.3, 4.1.2, 4.1.2, 5.2.2

[25] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*. De Gruyter Open, 2015. 1, 1.3, 5.2.1, 5.2.3

[26] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *PoPETs*, 2015(1):92–112, 2015. 1.3, 5.1.2

[27] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *Proceedings of 37th Symposium on Security and Privacy (Oakland 2016)*, pages 598–617, 2016. 1.1.2, 1.2.3, 3, 4.1.5

[28] Anupam Datta, Matt Fredrikson, Gihyuk Ko, Piotr Mardziel, and Shayak Sen. Use privacy in data-driven systems: Theory and experiments with machine learnt programs. Unpublished Manuscript, June 2017. 4

[29] Wendy Davis. Ftc's julie brill tells ad tech companies to improve privacy protections, 2016. URL `http://www.mediapost.com/publications/article/259210/ftcs-julie-brill-tells-ad-tech-companies-to-impro.html`. Accessed Nov 11, 2016. 1.2.1

[30] Dorothy E. Denning and Peter J. Denning. Certification of programs for secure information flow. *Commun. ACM*, 20(7):504–513, 1977. 1.1, 1.3, 5.1.1

[31] Henry DeYoung, Deepak Garg, Limin Jia, Dilsun Kaynar, and Anupam Datta. Experiences in the logical specification of the hipaa and glba privacy laws. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, pages 73–82, New York, NY, USA, 2010. ACM. 2.1

[32] Flávio du Pin Calmon and Nadia Fawaz. Privacy against statistical inference. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1401–1408, October 2012. URL `http://ieeexplore.ieee.org/abstract/document/6483382/`. 1.3, 5.2.1

[33] Charles Duhigg. How companies learn your secrets, 2012. URL `http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html`. Accessed Aug 13, 2016. 1.2.1, 4

[34] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS 2012)*, pages 214–226, 2012. 1.3, 4.2, 5.2.2

[35] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006. ISBN 3-540-35907-9. URL `http://dx.doi.org/10.1007/11787006_1`. 1.2.1

[36] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32731-2, 978-3-540-32731-8. 3

[37] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006. 1.3, 5.2.1

[38] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214–226, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1115-1. URL `http://doi.acm.org/10.1145/2090236.2090255`. 1.2.2

[39] European Commission. General data protection regulation (GDPR). Regulation (EU) 2016/679, L119, May 2016. 1

[40] Executive Office of the President. Big data: A report on algorithmic systems, opportunity, and civil rights. Posted at `https://www.whitehouse.gov/sites/default/files/microsites/ostp/2016_0504_data_discrimination.pdf`, May 2016. Accessed Oct. 17, 2016. 1

[41] Federal Committee on Statistical Methodology. Statistical disclosure limitation methodology. Statistical Policy Working Paper 22, 2005. 1.3, 5.2.1

[42] Federal Financial Institutions Examination Council. Home mortgage disclosure act data, 2011. URL `https://www.ffiec.gov/hmda/`. 1.2.3, 4.4

[43] Federal Reserve. *Consumer Compliance Handbook*, chapter Federal Fair Lending Regulations and Statutes: Overview. Federal Reserve, 2016. 4.3

[44] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 259–268, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3664-2. 1.2

[45] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 259–268, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3664-2. URL `http://doi.acm.org/10.1145/2783258.2783311`. 4.1.2, 4.1.2, 5.2.3

[46] Deepak Garg, Limin Jia, and Anupam Datta. Policy auditing over incomplete logs: theory, implementation and applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 151–162, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0948-6. doi: 10.1145/2046707.2046726. 5.1.1

[47] Deepak Garg, Limin Jia, and Anupam Datta. Policy auditing over incomplete logs: theory, implementation and applications. In *Proceedings of The ACM Conference on Computer and Communications Security (CCS)*, 2011. 4.2

[48] Google. Privacy policy. Accessed Nov. 21, 2014. 4

[49] Saikat Guha, Bin Cheng, and Paul Francis. Challenges in measuring online advertising systems. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 81–87, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0483-2. 5.1.2

[50] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003. ISSN 1532-4435. 5.1.2

[51] J.Y. Halpern and J. Pearl. Causes and explanations: A structural-model approach. part i: Causes. *The British journal for the philosophy of science*, 56(4):843–887, 2005. 5.1.2

[52] Katia Hayati and MartÃ■n Abadi. Language-based enforcement of privacy policies. In *In Proceedings of Privacy Enhancing Technologies Workshop (PET)*. Springer-Verlag, 2004.

5.1.1

[53] Xi He, Ashwin Machanavajjhala, and Bolin Ding. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2014)*. ACM, June 2014. URL `http://research.microsoft.com/apps/pubs/default.aspx?id=226369`. 1.3, 5.2.1

[54] Information Commissioner's Office, United Kingdom. Google inc.: Data protection audit report, 2011. URL `http://ico.org.uk/~/media/documents/disclosure_log/IRQ0405239b.ashx`. 1.1.1

[55] D. Janzing, D. Balduzzi, M. Grosse-Wentrup, and B. Schölkopf. Quantifying causal influences. *Ann. Statist.*, 41(5):2324–2358, 10 2013. 5.1.2

[56] T. Kamishima, S. Akaho, and J. Sakuma. Fairness-aware learning through regularization approach. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW 2011)*, pages 643–650, 2011. 1.2, 1.3, 5.2.2

[57] S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proceedings of the 49th IEEE Symposion on Foundations of Computer Science (FOCS 2008)*, pages 531–540, Oct 2008. 3

[58] Anthony Kennedy. Texas department of housing & community affairs v. the inclusive communities project, inc. Opinion of the United States Supreme Court, June 2015. 4.3

[59] Mathias Lécuyer, Guillaume Ducoffe, Francis Lan, Andrei Papancea, Theofilos Petsios, Riley Spahn, Augustin Chaintreau, and Roxana Geambasu. Xray: Enhancing the web's transparency with differential correlation. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, pages 49–64, Berkeley, CA, USA, 2014. USENIX Association. ISBN 978-1-931971-15-7. 5.1.2, 5.2.1

[60] Mathias Lecuyer, Riley Spahn, Yannis Spiliopolous, Augustin Chaintreau, Roxana Geambasu, and Daniel Hsu. Sunlight: Fine-grained targeting detection at scale with statistical confidence. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 554–566, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3832-5. 4.1.2, 4.1.2, 5.1.2, 5.2.1, 5.2.3

[61] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.*, 9(3):1350–1371, 09 2015. 5.3

[62] Peng Li and Steve Zdancewic. Downgrading policies and relaxed noninterference. In *POPL*, pages 158–170. ACM, 2005. ISBN 1-58113-830-X. 5.1.1

[63] M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`. 3

[64] Richard J. Lipton and Kenneth W. Regan. Making public information secret, 2016. URL `https://rjlipton.wordpress.com/2016/05/20/making-public-information-secret/`. Accessed Aug 13, 2016. 1.2.1, 5.2.1

[65] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar,

Matt Tolton, and Theo Vassilakis. Dremel: Interactive analysis of web-scale datasets. *PVLDB*, 3(1):330–339, 2010. 1.1.1, 2, 2.2

[66] Microsoft. Microsoft privacy statement, September 2016. URL `https://privacy.microsoft.com/en-us/privacystatement`. 4

[67] Tim Moses et al. Extensible access control markup language (xacml) version 2.0. *Oasis Standard*, 200502, 2005. 5.1.1

[68] Andrew C. Myers and Barbara Liskov. Protecting privacy using the decentralized label model. *ACM Trans. Softw. Eng. Methodol.*, 9(4):410–442, 2000. 1.3, 5.1.1

[69] Helen Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford Law Books. Stanford University Press, 2009. 4.2

[70] The President's Council of Advisors on Science and Technology. Big data and privacy: A technological perspective. Technical report, Executive Office of the President, May 2014. 1.2.1

[71] Office for Civil Rights. Summary of the HIPAA privacy rule. OCR Privacy Brief, U.S. Department of Health and Human Services, 2003. 4

[72] Parliament of Canada. Personal information protection and electronic documents act (PIPEDA). S.C. 2000, c. 5, 2000. 1

[73] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. ISBN 052189560X, 9780521895606. 5.1.2

[74] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 560–568, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. 1.2, 5.2.2

[75] François Pottier and Vincent Simonet. Information flow inference for ml. In *POPL*, pages 319–330, 2002. 1.3, 5.1.1

[76] Lewis F. Powell, Jr. Mcdonnell douglas corp. v. green. Opinion of the United States Supreme Court, May 1973. 4.3

[77] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, Berkeley, Calif., 1961. University of California Press. 5.1.2

[78] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why 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*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL `http://doi.acm.org/10.1145/2939672.2939778`. 5.3

[79] Stefan Rüping. *Learning interpretable models.* PhD thesis, Dortmund University of Technology, 2006. http://d-nb.info/997491736. 5.3

[80] A. Sabelfeld and A.C. Myers. Language-based information-flow security. *Selected Areas in Communications, IEEE Journal on*, 21(1):5–19, 2003. ISSN 0733-8716. doi: 10.1109/JSAC.2002.806121. 5.1.1

[81] Andrei Sabelfeld and David Sands. Declassification: Dimensions and principles. *Journal of Computer Security*, 17(5):517–548, 2009. 5.1.1

[82] Salman Salamatian, Amy Zhang, Flávio du Pin Calmon, Sandilya Bhamidipati, Nadia Fawaz, Branislav Kveton, Pedro Oliveira, and Nina Taft. Managing your private and public data: Bringing down inference attacks against your privacy. *J. Sel. Topics Signal Processing*, 9(7):1240–1255, 2015. URL `http://dx.doi.org/10.1109/JSTSP.2015.2442227`. 1.3, 5.2.1

[83] Shayak Sen, Saikat Guha, Anupam Datta, Sriram K. Rajamani, Janice Tsai, and Jeannette M. Wing. Bootstrapping privacy compliance in big data systems. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, SP '14, pages 327–342, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-4686-0. URL `http://dx.doi.org/10.1109/SP.2014.28`. 1.1, 1.2.3, 2, 4

[84] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. ISSN 1538-7305. 5.1.2

[85] G. Smith. Quantifying information flow using min-entropy. In *Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST 2011)*, pages 159–167, 2011. 1.3, 5.1.2

[86] Geoffrey Smith. Recent developments in quantitative information flow (invited tutorial). In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '15, pages 23–31, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4799-8875-4. URL `http://dx.doi.org/10.1109/LICS.2015.13`. 1.2

[87] Hugo Teufel III. Privacy policy guidance memorandum: The fair information practice principles: Framework for privacy policy at the Department of Homeland Security. Memorandum Number: 2008-01, December 2008. URL `https://www.dhs.gov/xlibrary/assets/privacy/privacy_policyguide_2008-01.pdf`. 1

[88] Ashish Thusoo, Joydeep S. Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu, and Raghotham Murthy. Hive - a petabyte scale data warehouse using Hadoop. In *ICDE '10: Proceedings of the 26th International Conference on Data Engineering*, pages 996–1005. IEEE, March 2010. ISBN 978-1-4244-5445-7. doi: 10.1109/icde.2010.5447738. 1.1.1, 2, 2.2

[89] J. Tian and J. Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1-4):287–313, 2000. 5.1.2

[90] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society Series B*, 73(3):273–282, 2011. 5.3

[91] Florian Tramèr, Vaggelis Atlidakis, Roxana Geambasu, Daniel J. Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. Discovering unwarranted associa-

tions in data-driven applications with the fairtest testing toolkit. *CoRR*, abs/1510.02377, 2015. URL `http://arxiv.org/abs/1510.02377`. 1.2, 4.1.2, 4.1.2, 4.1.5, 5.2.3, 5.2.3

[92] Michael Carl Tschantz and Jeannette M. Wing. Extracting conditional confidentiality policies. In *Proceedings of the 2008 Sixth IEEE International Conference on Software Engineering and Formal Methods*, pages 107–116, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3437-4. doi: 10.1109/SEFM.2008.46. 5.1.1

[93] Michael Carl Tschantz, Anupam Datta, and Jeannette M. Wing. Formalizing and enforcing purpose restrictions in privacy policies. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pages 176–190, Washington, DC, USA, 2012. 1.3, 4, 5.2.1

[94] Michael Carl Tschantz, Anupam Datta, and Jeannette M. Wing. Purpose restrictions on information use. In *Computer Security - ESORICS 2013*, volume 8134 of *Lecture Notes in Computer Science*, pages 610–627. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40202-9. doi: 10.1007/978-3-642-40203-6_34. 5.1.1

[95] Michael Carl Tschantz, Anupam Datta, and Jeannette M. Wing. Purpose restrictions on information use. In *Proceedings of the 18th European Symposium on Research in Computer Security (ESORICS)*, volume 8134 of *Lecture Notes in Computer Science*, pages 610–627. Springer Berlin Heidelberg, 2013. 4

[96] Michael Carl Tschantz, Amit Datta, Anupam Datta, and Jeannette M. Wing. A methodology for information flow experiments. In *Computer Security Foundations Symposium*. IEEE, 2015. 1

[97] Joseph Turow. *The Daily You: How the New Advertising Industry Is Defining Your Identity and Your Worth*. Yale University Press, 2011. ISBN 9780300165012. 1.2.1

[98] Findings under the Personal Information Protection and Electronic Documents Act (PIPEDA). Use of sensitive health information for targeting of google ads raises privacy concerns, 2014. URL `https://www.priv.gc.ca/cf-dc/2014/2014_001_0114_e.asp`. Accessed Aug 13, 2016. 1.2.1

[99] U.S. Federal Goverment. Part 1607—uniform guidelines on employee selection procedures. Code of Federal Regulations, Title 29 - Labor, Vol. 4, 1978. URL `https://www.gpo.gov/fdsys/pkg/CFR-2011-title29-vol4/xml/CFR-2011-title29-vol4-part1607.xml`. 4.3

[100] Berk Ustun, Stefano TracÃă, and Cynthia Rudin. Supersparse linear integer models for interpretable classification. *ArXiv e-prints*, 2013. URL `http://arxiv.org/pdf/1306.5860v1`. 5.3

[101] Jeffrey A. Vaughan and Stephen Chong. Inference of expressive declassification policies. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, pages 180–195, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4402-1. doi: 10.1109/SP.2011.20. 5.1.1

[102] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*,

pages 325–333, 2013. 1.2, 1.3, 5.2.2