Author version of the paper published at 39th International Conference on Distributed Computing Systems, ICDCS, 2019, Copyright: IEEE

The Best of both Worlds: Challenges in Linking Provenance and Explainability in Distributed Machine Learning

Stefanie Scherzinger*
OTH Regensburg
Regensburg, Germany
stefanie.scherzinger@oth-regensburg.de

Christin Seifert*
University of Twente
Enschede, Netherlands
c.seifert@utwente.nl

Lena Wiese*
Leibniz University Hannover
Hannover, Germany
lena.wiese@udo.edu

***Abstract*—Machine learning experts prefer to think of their input as a single, homogeneous, and consistent data set. However, when analyzing large volumes of data, the entire data set may not be manageable on a single server, but must be stored on a distributed file system instead. Moreover, with the pressing demand to deliver *explainable* models, the experts may no longer focus on the machine learning algorithms in isolation, but must take into account the distributed nature of the data stored, as well as the impact of any data pre-processing steps upstream in their data analysis pipeline. In this paper, we make the point that even basic transformations during data preparation can impact the model learned, and that this is exacerbated in a distributed setting. We then sketch our vision of end-to-end explainability of the model learned, taking the pre-processing into account. In particular, we point out the potentials of linking the contributions of research on data provenance with the efforts on explainability in machine learning. In doing so, we highlight pitfalls we may experience in a distributed system on the way to generating more holistic explanations for our machine learning models.**

***Index Terms*—Provenance, explainable machine learning, distributed computing.**

I. Introduction

It's the era of Big Data: The amount of digitally available data is growing exponentially, data is becoming more and more diverse. This is mainly attributable to a growing amount of user-generated content (e.g. blogs) and smart, data-generating devices (Internet of Things). Even with the increasing computational power and memory, analyses of these datasets can not be performed on single machines anymore. In distributed computing, or more specifically distributed machine learning, huge machine learning models are computed in parallel (i.e., model parallelization), or on distributed data (data parallelization) [1]. At the same time, the demand for fair, explainable and accountable machine learning increases, due to automated decision making in sensitive domains such as health care, and due to legal requirements [2], [3].

Deep Learning Models, the de-facto state-of-the-art models in machine learning, already achieve near-human performance on many tasks [4], while at the same time being inherently complex. Their complexity originates from their sheer size (e.g., the VGG-19 net for image recognition has 19 layers and ≈143 million parameters [5]), the use of non-linear functions (e.g. subsequent application of convolutions, pooling and ReLU activations in computer vision) and potentially recursive or recurrent internal processing (e.g., in recurrent neural networks). Due to this complexity, humans need novel mechanisms to understand the machine learning model and appropriately interpret the decisions that these models derive. The topic of explainability in machine learning focuses on such mechanisms.

In order for explanations to truthfully explain how an automated ML approach arrived at a decision, *all* transformations and functions that have been applied to the data have to be considered. This includes data cleaning and preprocessing steps as well as means for distributed processing (e.g. sampling for training sets of different models), that need to be considered by the ML explanation. Yet as of today, this is not the case. However, from the viewpoint of data management and data preparation, the database community already provides means for tracking transformations applied to raw data. There is a large body of work on tracing *data provenance* (or *data lineage* [6]) throughout the data preparation pipeline. The goal is to account for the origin of a record together with a trace of how and why it got to the present place.

In this paper, we take the following viewpoint: instead of considering explainability or provenance in isolation, or the legal issue of accountability, we aim at full explainability of automated decision making in distributed systems. Accordingly, we outline challenges that arise when trying to integrate the notions of explainability and provenance. We argue that provenance information is crucial for developing reliable and trustworthy explanations of (distributed) machine learning models, which we call end-to-end explanations. We argue in favor of joint research and the development of a novel paradigm for true end-to-end explanations. In particular, we

- argue, along a specific example, how data preparation and distributed processing can affect machine learning models and their explanations,
- sketch where provenance information should be collected and how it can be exploited for reliable explanations of

[*] All authors contributed equally

distributed machine learning models, and

- lay out a range of nontrivial research challenges to be mastered for reliable and trustworthy explanations of machine learning models.

Structure: We next provide some background on key concepts used throughout this paper, such as the term *provenance*. We then walk through a small example use case of distributed machine learning in Section III and discuss how provenance information can be of use. The identified challenges for end-to-end explanations are then generalized in Section IV. The central idea of this paper is to leverage the concepts of provenance from database research and apply them to obtain trustworthy explanations of machine learning models; related work on provenance and explanations is discussed in Section V. We then conclude the paper.

II. Background

This section introduces background terminology for explainability in machine learning (ML) and provenance in databases.

A. Explainability and Model Provenance

Explainability in ML has so far addressed different notions of explanations. Explanations for different machine learning models either

- show instances that are similar and lead to the same decision (case-based explanations, e.g. [7]),
- outline differences in otherwise similar cases (counterfactual explanations, e.g., [8]),
- show how much a feature contributes to the final decision (feature importance based explanations, e.g., [9]), or
- use/convert to a inherently understandable model (model-based explanations, e.g., [10]).

We will illustrate these notions in our upcoming example.

In this article, the term *model provenance* means the tracking of all metadata (such as configurations, random number seeds, hyperparameters) needed to reproduce the training of the machine learning model (similar to the metadata tracking for machine learning algorithms in [11]).

B. Data Provenance

The terms data provenance and lineage are often used interchangeably, yet sometimes the former is used to only refer to the point of origin of a data record, while the latter includes the transformation process. In this paper, we use them synonymously. Data provenance is well-studied in both theoretical and systems research [12], [13]. In fact, there are publications dating back as long as 15 years [14], and again very recently in the context of data science [15], stressing the importance of tracing provenance throughout the data preparation pipeline so that it may be leveraged in succeeding stages. Keeping track of data flows and decisions or actions taken based on data has been identified as key for accountable data-driven interconnected decision systems [16]. This concept was further called decision provenance, as a term for system accountability, by the authors of [16].

Provenance in databases can be further categorized into (i) *where provenance*, addressing the origin of tuples from a table/attribute perspective, (ii) *how provenance*, addressing the nature of the processing steps applied to tuples, (iii) *why provenance*, addressing any "source tuples" involved in the derivation, and (iv) *why-not provenance*, addressing the reasons why tuples are missing from the result.

Related research directions include the explainability of SQL queries, i.e. identifying the part of the provenance relevant for answering a query [17], answering *what-if* questions for hypothetical query execution [18], and SQL query explanation and debugging [19]. The latter also involves provenance tracing, with the narrow goal to explain query behavior.

Research on data provenance has produced a principled theory of commutative semirings [20], allowing to propagate provenance information through query evaluation.

III. Example Scenario

In this section we discuss a small example on a toy data set to illustrate

- the merits of generating better explanations using provenance information, and
- the challenge of producing both the provenance information and the explanations in a distributed setting.

We envision a scenario where we learn a machine learning model from data too large to be stored on a single machine. In order to illustrate the challenges of distributed machine learning in this paper, we choose an ensemble of decision trees as machine learning model. Ensemble methods aggregate decisions made by so-called base classifiers [21]. Base classifiers are either different models from the same model class trained on the same data chunk, or models from different model classes with different capabilities. The idea is that each base classifier is an expert on some aspect in the data and the final decision is based on an aggregation of expert opinions. We choose decision trees as base classifiers because of their inherent interpretability and straight-forward visualization[1].

As the aggregation method, we choose averaging of class a-posteriori probabilities and maximum a-posteriori, such that from each decision tree we receive a probability for each class, average those over all decision trees, and choose the class with the maximum value as the final decision[2]. An example of such an ensemble is shown in Figure 1.

In this example, distributed machine learning is approached by i) computing one decision tree for each chunk of data (distributed across the computational nodes) and ii) aggregating their output into a final decision. To summarize,

1) we distribute the training data across the nodes in a distributed file system, such as HDFS;
2) locally, each node cleans its data regarding missing and noisy values; each node computes its own decision tree;

[1] With a decision tree model, it is easy to determine which feature combinations lead to a specific decision and – if the decision tree is small enough – the whole model can be translated into a small batch of if-then rules, as also discussed in Example 3.

[2] This procedure differs from Random Forest classifiers, since the base classifiers are generated on informative and not on random features.

TABLE I
RAW DATA.

Name (N)	Age (A)	Pizzas (P)	Sport (S)	Fit (F)
TRAINING DATA				
Amy	35	0	1	1
Bob	20	2	1	1
Charlie	32	2	0	0
Dave	*null*	5	*null*	`"N"`
Eve	24	*null*	1	`"0"`
Francis	35	0	1	1
Greg	20	0	1	1
Haley	32	2	0	0
TEST DATA				
Zoe	40	7	1	?

TABLE II
PARTITIONED TRAINING DATA FROM TABLE I, AFTER LOCAL DATA CLEANING AND VALUE IMPUTATION. ATTRIBUTES NAMES ARE ABBREVIATED.

N	A	P	S	F
A	35.	0.	1.	1
B	20.	2.	1.	1
C	32.	2.	0.	0
D	23.04	5.	0.68	0
E	24.	2.94	1.	0

N	A	P	S	F
F	35.	0.	1.	1
G	20.	4.	1.	1
H	32.	2.	0.	0

3) the machine learning expert runs an algorithm to aggregate the locally computed trees into an ensemble, and publishes it to the end users as the resulting model;
4) ultimately, the end users apply the final ensemble model to classify new records.

We will outline how provenance data can be valuable for performing the last two steps, and accordingly, which provenance information has to be tracked in the first two steps.

A. The Running Example

Suppose want to predict the fitness of adults, depending on their pizza consumption and exercise habits. Table I shows the raw data for patients Amy through Haley. Physicians have collected data on their patients' names, their age, the number of pizzas they eat per week on average, and whether they exercise at all. Further, they have determined their general level of fitness, denoted by 0 (unfit) and 1 (fit). Some values are missing (denoted *null*), and some values are noisy.

Zoe is a new patient. Her fitness should be predicted based on her other attributes and the data from the other patients.

Table II shows the data distributed across two compute nodes: data are split into two disjoint partitions, that will be processed independently. After data partitioning, a local data preparation step was performed: The nodes have repaired inconsistent entries in the *Fit* column by translating `"N"` and `"0"` to 0. Since decision tree learners usually cannot handle missing values, a nonparametric estimator is used to impute the *null* values based on the observed other values for each feature [22]. (Incidentally, no data was changed during this step in the second partition.)

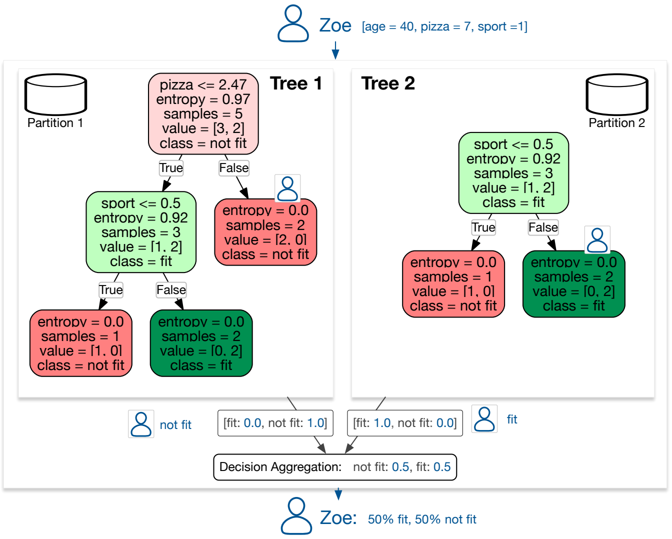

Fig. 1. Ensemble of decision trees. Trees 1 and 2 have been trained locally, on the distributed data partitions from Table II. When classifying test patient Zoe, the locally trained trees do not agree.

Figure 1 shows the decision trees 1 and 2 computed locally from both data partitions.[3] In tree 2, the only feature relevant for predicting fitness is whether a person exercises regularly. Based on this model, the reasonable (and only) medical intervention would be to recommend regular exercise. While this is surely not wrong, it only reflects part of the real world. Decision tree 1 relies both on nutrition and exercise as informative features. Intervention based on this decision tree is to increase exercise, as well as to reduce pizza intake.

Now, let us consider Zoe, age 40, daily consumer of pizzas and intensive exerciser. Interestingly, Zoe is classified as unfit by tree 1, and fit by tree 2. The best decision the ensemble of trees can make, after aggregating the decisions of the single trees, is to state that Zoe is just as likely fit as she is unfit.

B. Provenance for ML Experts

Typically, the basis for tracing provenance are annotations to the data, as illustrated next.

Example 1: Table III again shows the data from the first partition after local pre-processing. The last two columns contain provenance annotations for each tuple. The penultimate column (in a notational style inspired by [23]) describes how each entry was derived, using relational algebra operators.

[3] All trees shown were generated in python using the sklearn library. The Jupyter notebook and the data are available online at https://bit.ly/2MYtRal. The visualization encodes the following information: The base color of the node represents the classification made for data records that pass trough this node, green for "fit" and red for "not fit". The saturation in color represents the confidence of this decision, highly saturated means highly confident. The first line of text labelling a node indicates the splitting attribute and the decision threshold (e.g. pizza ≤ 2.47), the last line is the decision made in this node. The label entropy indicates the splitting criterion (here, information gain) and the value of this criterion in this specific node. Further shown are the number of training samples passing through the node and their respective distribution over the classes of interest. "value = [3,2]" in the root of tree 1 means there are five samples in total, three from class "not fit" and two from class "fit".

TABLE III

DATA PROVENANCE FOR THE FIRST DATA PARTITION AS ANNOTATIONS ON TUPLES.

N	A	P	S	F	directly derivable from the raw data in table T_1 by...	conf.
A	35.	0.	1.	1	$T_1(A, 35, 0, 1, 1)$	100%
B	20.	2.	1.	1	$T_1(B, 20, 2, 1, 1)$	100%
C	32.	2.	0.	0	$T_1(C, 32, 2, 0, 0)$	100%
D	23.04	5.	0.68	0	$t := T_1(D, null, 5, null, \text{“}N\text{”});$ $\pi_{N,P}(t) \times (A : \mathit{Imp}_A(T_1, t)) \times (S : \mathit{Imp}_S(T_1, t)) \times \pi_F\big(\sigma_{F'=\text{“}N\text{”} \vee F'=\text{“}0\text{”}}(\rho_{F' \leftarrow F}(t) \times (F : 0))\big)$	60%
E	24.	2.94	1.	0	$t := T_1(E, 24, null, 1, \text{“}0\text{”});$ $\pi_{N,A,S}(t) \times (P : \mathit{Imp}_P(T_1, t)) \times \pi_F\big(\sigma_{F'=\text{“}N\text{”} \vee F'=\text{“}0\text{”}}(\rho_{F' \leftarrow F}(t) \times (F : 0))\big)$	60%

By T_1, we refer to the table holding the raw data assigned to the first partition. For instance, the record for Amy is unchanged, but the records for Dave and Eve differ from the raw data. The function $\mathit{Imp}_A(T_1, t)$ imputes the missing age value in tuple t, based on the locally available table T_1; we proceed similarly with the other imputation functions.

The last column records modifications to tuples on a much more coarse-grained level, stating the confidence in this tuple as the percentage of attributes that remain unchanged. This idea of annotating confidence values is highly related to the works on probabilistic databases, c.f. [24], and techniques for propagating such confidence values through query evaluation. □

We now picture the machine learning expert evaluating the trained model:

1) Considering Data Uncertainty: Having requested how and where provenance, the machine learning expert obtains a trail that accounts for the origin of a record, together with an explanation of how and why it got to the present place.

Example 2: Having classified the test record from patient Zoe, our expert finds out that the final a-posteriori distribution is uniform and the model cannot make a confident prediction for any of the classes. Our expert starts with requesting how and where provenance on the input data for tree 1.

She could then be presented with fine-grained data provenance: references to the raw tuples, in addition to the transformations applied, as seen in Table III. While these annotations exactly record the processing steps applied to each raw tuple, and allow a *white box* view, our machine learning expert may not be familiar enough with database theory so as to be fluent in relational algebra. Moreover, this low-level, per-tuple annotation may be simply too much information to be explored manually, given that our expert is processing Big Data.

We therefore also consider an alternative: The last column contains coarse-grained provenance information called *confidence*, merely stating the similarity of a tuple compared to its original. This can be considered a *grey box* view of preprocessing (allowing more insight than treating the model as a *black box*). At least, our expert can now tell that some tuples have been modified, since not all input tuples have 100% confidence. Naturally, confidence might also be tracked on the level of single values, not just entire tuples. □

We defer the extended discussion of finding an appropriate level of granularity in tracking provenance to Section IV, and continue with sketches of how the machine learning expert may leverage her provenance-triggered insights. On an aggregated level, she finds out that average data confidence for the first partition amounts to 84%, while it contains 63% of the tuples. The other partition contains 37% of the tuples with 100% data confidence.

It is now up to the expert whether to account for the data uncertainty in building her model, or whether to even exclude problematic tuples from the analysis altogether. Considering data uncertainty, she could conclude that while the first decision tree is built on a larger share of the data (63% of all tuples), and thus more trustworthy, it also has higher data uncertainty (84% certainty on average). A simple approach would be to not consider the output of each tree equally, but rather weight the decision from the tree 1 by the factor $0.63 \cdot 0.84$, and the decision from tree 2 by the factor 0.37.[4]

2) Bias, Skewness, and Fairness: Upon visual inspection of the patient names, it seems that the raw data from Table I predominantly describe male patients (this information is not directly encoded, but only deducible from the persons' names). In our specific scenario, decisions might need to be gender-sensitive. Moreover, the data distribution according to Table II produces skewed data: The first partition is dominated by unfit patients, whereas the second partition is dominated by fit patients, all male. Dealing with biased and skewed data is an active issue in machine learning research, as we point out in our discussion of related work in Section V.

The fact that the data itself is biased towards gender is not so straightforward to detect automatically. However, the skewness of the data partitioning may be discovered, for instance, by comparing summary statistics from the global data set against the locally available data set. One straightforward option is to compare histograms, which can be efficiently computed in a distributed fashion, e.g. using MapReduce [25], [26].

In fact, we consider it quite likely that the data available on a local node within a distributed file system, such as HDFS, is skewed: Data warehousing or data lake scenarios commonly follow the "write once, read many times" paradigm: Incoming data is chopped into HDFS chunks and distributed by HDFS in the system. Over time, new chunks are added, yet existing

[4] This weighting results in scores that are not a probability distribution over classes anymore. Normalisation might be added if class probabilities are required for the final output.

chunks are never updated.[5] Consequently, when the global distribution of data changes over time, this is not reflected in older chunks.

As we also discuss in Section IV, we believe that this challenge may be addressed systematically, by extending distributed machine learning algorithms such that they account for skewness. In spirit, this is related to the efforts in the machine learning algorithms to be subgroup-fair, discussed in Section V, or more generally, to develop machine learning algorithms that are bias-aware.

Additionally, we may even envision new and sophisticated data placement strategies for data chunks, beyond the HDFS strategy of sharding chunks across the physical nodes such as to evenly distribute the workload. In general, devising dynamic data placement strategies to improve system performance already is a current field of research [28]–[32]. With data skewness in mind, chunks might also be placed such that due to co-location of complementary chunks, the probability of a physical node working with skewed data is reduced.

C. Provenance for End Users

In the following, we envision patient Zoe using the computed classifier as a *black box model* to get an estimate of her fitness during her annual medical checkup. We walk through several scenarios where the explainability of her result can be improved based on provenance information, thus providing a *grey box model*.

As we also discuss as part of the research challenges, it remains an open question in what form provenance information can be integrated with this black box model, to be made available to the end user.

1) Model-based Explanations: Model-based explanations attempt to generate a white-box view of the black box (not understandable) machine learning model. For our base classifiers, the decision trees, we derive an explanation by simply converting each decision tree into a rule. For small decision trees, this rule is comprehensive and easily understandable by humans. The explanation for an ensemble is more involved and depends on the aggregation method used. Additionally, general information about training the algorithm (that is, *model provenance*) can be provided in compiling an explanation. For instance, this could be the size of the training set, hyperparameters (i.e. the configuration parameters of the machine learning algorithm), and the performance of the trained model on a separate data set that has not been used for training.

Example 3: Tree 1 from Figure 1 translates to the rule

```
if pizza ≤ 2.47 and sport > 0.5 then fit,
```

and tree 2 to

```
if sport > 0.5 then fit.
```

Hyperparameters include the splitting criterion (here: information gain) and the minimum samples required for a node to attempt a new split (two in this example). The decision of the final ensemble can then be explained as taking the average of the previous decision values. Showing this to the end-user is not straight-forward, however, our aggregation function is a linear function of the output of the decision trees and linear functions are considered understandable [10].

If provenance information is available, the explanation can reflect that there was uncertainty in the input data because of missing values and that the decision is biased towards males, resulting in more holistic and truthful explanations. □

2) Feature-based Explanations: Feature-based explanations highlight the features that contributed most to the final decision. Feature-based explanations are straightforward to extract from decision-trees, by simply tracing the path a data record took from the root of the tree to the leaf.

Example 4: In our example (Fig. 1), a feature-based explanation for the decisions of tree 2 would state that sport has a positive influence on fitness, while for tree 1, additionally, pizza has a negative (and stronger) influence. Both are valid explanations that capture the decisions made by their respective model. Yet the first does not truthfully reflect the real world[6]. The feature-based explanation for the final decision of the ensemble is then the list of features from the trees ranked according to their importance in the single trees, and – if provenance information were available – weighted by the trustworthiness of each tree. The trustworthiness value would be calculated based on considerations about data uncertainty and potential biases of the model. □

3) Case-based Explanations: Case-based explanations show records that are treated as similar by the algorithm, such as a group of (anonymized) patients for whom the same decision was made.

Example 5: Zoe (age 40, avid fan of pizzas and exercise, classified as unfit by tree 1, fit by tree 2, and undecidable by the tree ensemble), requests case-based explanations. Because the final decision is equally based on tree 1 and tree 2, and for the sake of conciseness, we focus our discussion on obtaining case-based explanations for single trees.

- Given tree 1, the system would list Dave and Eve who eat more than 2.47 pizzas per week, and despite exercising regularly, are nevertheless unfit.
- Given tree 2, the system would list Francis and Greg as similar, since they also exercise and are also fit.

Given the information from tree 1, Zoe might again realize that while Dave and Eve do exercise and eat a lot of pizza, the sports value of Dave (0.68) is only a guess. This raises the question whether this decision can be fully trusted.

To dig deeper, our Zoe (or her general practitioner) might state a case-based "what-if" question: who is similar to Zoe in terms of behaviour, but is classified as fit instead of unfit? As answer, Bob will be listed, showing that a moderate pizza consumption in combination with exercise indicates fitness, which seems reasonable and also actionable for Zoe. □

[5] In fact, early versions of database systems like Hive, built on top of HDFS technology, did not even provide any means for updating data, c.f. [27].

[6] In fact, both models are abstractions of the real world and do not completely reflect it, but tree 1 is a more complete representation

IV. Challenges and Ideas for Solutions

Achieving end-to-end explainability for machine learning in distributed systems comes with its own challenges. We next categorize these challenges and sketch first ideas.

A. Access to Provenance Information

For generating truthful explanations, we would like to guarantee that all data processing steps are repeatable [13], and we also have all information on *model provenance*, i.e., the training of the ML model. In consequence, then the model and all its predictions are reproducible [11].

Unfortunately, most of the popular machine learning libraries do not make their internals transparent. For instance, the scikit-learn library used in our example in Section III to impute missing values does not provide any programmatic access to meta information on how this imputation works. In order to track what-provenance for the values that have been imputed, we would need to extract the necessary information from the publication associated with the machine learning library, and ideally (or in case no publication is linked to the library), inspect (and understand) the source code of its implementation. Otherwise, it is not clear whether the algorithm derives a missing value given all values in that relation's column, or whether it considers the complete relation.

Similarly, many ML algorithms rely on random choices and random data sampling. Random forests for instance, select the splitting features randomly [33][7], while neural networks randomly initialize all weighs between layers. For these cases, we might like to record even the seeds to the random number generators as part of data provenance. Yet virtually none of the popular machine learning libraries and tools make this data accessible. Similarly to the metadata tracked for reproducible experiments with algorithm developers as stakeholders [11], we need *model provenance* to fully explain the faithfulness of a machine learning model to end-users. Note that the model itself, i.e., the algorithm class and the trained model parameters, is not included in model provenance. The model itself is stored for predictions separately, but additional information is necessary to judge the usefulness of those predictions.

Solution Ideas: Making provenance information available in all data processing steps, whether pre-processing or model learning, requires the joint effort of both the database and the machine learning community. While this may seem a daunting engineering effort, it is a long-term investment for all communities involved. Besides faithful explanations of machine learning models, we further have the added benefit of reproducible machine learning workflows [11].

B. Provenance Granularity

One classic, open question is how verbose provenance data can be to be consumable, and at the same time, usable by a machine learning algorithm. Dealing with Big Data, in distributed systems, lends a new urgency to addressing this question. Coarse-grained provenance is merely a high-level description of the basic data preparation workflow and its stages. Yet fine-grained provenance means tracking at the granularity of individual records.

Data distribution exacerbates the granularity problem: each server will first of all track their local provenance independently – and they might have different requirements with respect to granularity. We can see this in our example as follows. The left partition in Table II is modified by preprocessing (imputation of values). When later on inspecting provenance information, we need a fine-grained information at cell level: we have to expose the exact way how the imputed cell values were derived – hence, for the imputed cells we have to maintain the information on which other values they are based. In contrast, the right partition in Table II remains unchanged during pre-processing. When requesting provenance information for this partition, tuple-level provenance is sufficient: we just have to provide the information that the original raw data were used in the machine learning pipeline.

Solution ideas: Future provenance systems should provide a feature of customized abstraction: Provenance information should be provided at a granularity matching the requirements of the machine learning method at hand. We therefore need mechanisms that can find out which granularity of provenance data is processable by the chosen machine learning method. In Example 2, we can see that the more abstract notion of confidence can be exploited by the decision tree learners later on to assess the quality of the models learned on distributed data. Yet, other methods could also exploit provenance tracking at the level of relational algebra expressions to achieve more truthful explanations. For programmatic access, an intelligent provenance system should be able to negotiate the granularity of provenance information between the data preparation layer and the machine learning tool. Moreover, as shown by the imputation example, potentially different levels of granularity should also be supported.

C. Data Volume

Building scalable solutions for storing and processing large volumes of provenance data is already an issue today. Yet when we combine the collection of provenance data with the quest for explainable machine learning algorithms, we will generate even more provenance data. Moreover, in a distributed setting, the provenance information will itself be distributed across compute nodes: Each server is responsible for tracking provenance data locally.

Solution Ideas: Future provenance systems need data structures and mechanisms to query distributed fine-grained provenance information efficiently. Thus, we may need to apply highly specialized, compressed data formats (e.g. multi-index data formats, such as HDF5).

D. Bias and Fairness

In distributed file systems, the focus is to optimally distribute the data to optimize access time and database operations. Consequently, the data on single machines might not

[7] Given the multitude of variations for the general Random Forest algorithm, the problem of computing provenance becomes even more pronounced.

be representative for the general data distribution (e.g., approx. 50% males and 50% females). This might lead to effects where the data trends detected on single machines differ from the general trend, and effect known as Simpson's paradox [34], or the resulting machine learning model might be highly biased and provide unfair decisions to certain subgroups. In Example 2, the second partition in Table II is highly biased towards males making the trained model prone to predict male fitness with much higher accuracy than female fitness [35]. In traditional single-machine machine learning, the data is contained in one data chunk and the learner uses the whole data set of random samples to ensure that the training data is representative of the whole data set. In distributed machine learning, random sampling across the whole data set is highly inefficient or not computationally feasible at all.

Solution Ideas: Future provenance systems should be integrated with novel bias-aware algorithms in machine learning; these should then take the underlying data distribution into account. For instance, in Bayesian classification, the class prior probability can first be calculated on single machines, then aggregated to the global class prior probability, which is shared with all workers, that then compute the posterior probability based on the locally available data. While this approach is straight-forward for Bayesian models, other classes of machine learning algorithms would need to be adapted. In addition, by obtaining statistics about the distributed partitions, biased data partitions can be identified and exposed.

Ideally, a better and less biased data distribution can be induced based on statistics, where the distributed file system assigns data chunks to physical nodes such as to take imbalances of the data distributions into account. Yet, in practice this might now always be possible due to conflicts with privacy requirements (see Section IV-H).

E. Provenance Visualization

An intrinsic challenge – even in single-node settings – is how to visualize provenance data in a consumable form to either machine learning experts or end users. In particular, for large data sets, it will be unfeasible to inspect every single raw record, heavily impairing the usability of the captured data [16]. Moreover, in a distributed setting, provenance visualization has to aggregate information from several servers.

Solution Ideas: Future provenance systems should be able to summarize data and model provenance traces e.g., by visualization techniques [36] allowing flexible access to different levels-of-detail [37]. Both the Human-Computer Interaction Community and the Visual Analytics community have already identified explainable, accountable and intelligible systems in general as part of their research agenda [38], [39]. With provenance information being crucial for trustworthy explanations, novel approaches need creative solutions collaboratively designed by machine learners, database experts, as well as the information visualization and human-computer interaction communities.

F. Data Freshness

In an advanced machine learning setting, a model might also consider the time at which a data item was created or modified; more recent data might get more influence when training the model. Explainability of ML models should also exploit the time of origin and modification in order to let the user assess whether the result is based on timely and fresh data. In a Big Data setting, where the machine learning expert relies on data provided by several independent sources, stale data might occur quite often and tracking data freshness is a more difficult problem. This also raises the question at which level of granularity the timestamping of data should take place.

Solution Ideas: Future provenance systems should be able to track the time of creation and modification of data and expose it to the machine learning algorithm in order to make the time information usable by explanations. In a distributed setting, a common notion of time has to be established, especially for application scenarios that require real-time decision making and/or are based on online learning.

G. Variability and Lack of Standards

It is not clear exactly which kind of provenance information is relevant for the machine learning community, as this problem space is huge: Academic research has produced hundreds of different machine learning algorithms, each with different requirements and abilities with respect to input data. Moreover, there are several competing notions of machine learning explainability. Further, the requirements on explanations for different stakeholders differ, and it is not clear what makes a good explanation. Most of the time, data pre-processing implementations or machine learning implementations are black boxes that do not expose enough information to be immediately exploitable for explanations; in our example, the exact ways how the imputed values were derived are not revealed when using the python library. Similarly, on the database side, we have competing notions of data provenance, as well as vast choice of database systems beyond those supporting the relational model [40]. Novel data models in return require customized provenance mechanisms.

Solution Ideas: Both communities need to consolidate, standardize, and agree on data exchange formats. To avoid technological lock-in, we need programming APIs and standards that let us to switch between database backends, provenance formalisms, and machine learning algorithms. In particular, in our distributed setting it might even be possible that different data sources use different database systems; hence provenance should also be supported by data integration systems and distributed query engines (like for example, Presto or Dremio).

H. Data Protection and Privacy

Compliance with data protection regulations is a major requirement for IT systems managing sensitive data. This in particular applies when integrating data from different data providers. For example, we can consider the case that the two partitions in Table II are independently managed by two different hospitals. These hospitals do not want to disclose the data of their patients. When each hospital removes all personally identifiable information (performing anonymization and pseudonymization) from the training data, the quality of the derived explanations is likely suffer, because external users of the model (e.g. Zoe and her physician) are not allowed to drill into the provenance information down to the level of the raw input tuples.

Solution Ideas: This calls for privacy-compliant tradeoffs between guaranteeing anonymization and yet providing provenance data and ML explainability. Existing formalisms like for example k-anonymity [41] and separation of duties [42] must be extended appropriately. In addition, novel cryptographic approaches (for example based on homomorphic encryption or secure multiparty computation) have to be developed to obtain provenance data without breaching privacy.

V. Related Work

Research on scaling up machine learning algorithms looks back at a long tradition [43]. With the recent proliferation of Big Data, interest in this field has been revived. In order to understand challenges and solutions for trustworthy explanations of automatic decision making systems in distributed settings, we review work in two partial aspects. First, we consider provenance for automated decision making. Next, we consider provenance research in distributed settings. We further discuss work on explanations of machine learning models, and bias and fairness in machine learning – a problem we identify as being more pronounced in distributed decision systems.

a) Provenance for Automated Decision Making: The idea of harvesting provenance information for aspects of automated decision making has been introduced before [11], [16]. Decision provenance aims at tracking data flows to enable the identification of entities responsible for a particular decision or action [16]. The goal of decision provenance is to ensure (legally) accountability of decision-making systems. Accountability and transparency are different from explainability: the latter focuses on *what was happening* for a complex decision to come into being, the former focuses on *who is responsible* for a particular (sub-)decision. Schelter et al. [11] introduce a data model for tracking metadata of machine learning models, such as model identifiers, model versions, hyperparameters, and identifiers of the training and test data used in the experiment. Their idea of using provenance data for machine learning models is similar to ours, however, their target users are algorithm designers and the goal of the framework is to assist model development. In this paper, we argue for data and model provenance to explain machine learning models also to end users and discuss associated challenges.

However, the machine learning community does not yet systematically exploit provenance information collected during data preparation. While there is a proposal for queryable provenance [44], there are no standardized interfaces yet. Making provenance information programmatically accessible is a prerequisite towards machine learning algorithms actually leveraging this information when deriving explanations.

b) Provenance in Distributed Settings: Naturally, the notion of data provenance tracking has long since been extended to MapReduce-based operators, e.g., in the RAMP framework [45]. Likewise, similar notions have been proposed for Spark transformations, both in academic research (e.g. [46]), and provenance tracking has by now been implemented within the official Apache Spark project itself [47].

Database provenance techniques have been successfully applied to other domains and different purposes. For instance, [48] applies them to diagnose problems in distributed systems.

c) Explanations for Machine Learning Models: While some research already addressed the understandability of machine learning models as early as 2000, (e.g. [49]), a new line of research on interpretable, explainable, trustworthy and fair algorithms started approximately in 2016, most prominently with the DARPA Explainable Artificial Intelligence (XAI) Programme and the EU General Data Protection regulation [3]. Explanation approaches of machine learning models address different stakeholders [50] and explain different aspects of the model. Three general approaches have emerged towards providing explanations [51]. First, explanations can be model-based by showing the operational procedure of the whole model. For some models, those explanations are easy to derive, e.g., a decision tree can be explained by either translating it to if-then rules or visualizing the tree. Complex models can be approximated with simpler ones, either locally (e.g., [10]) or globally (e.g., [52]). Global approximation by an inherently understandable model consequently explains the approximated model itself [53], fostering a global, general understanding [54]. Second, explanations can be case-based, showing data items that lead to the same decision (e.g., [7]). Third, explanations can be based on features, outlining which feature contributes to which extent to the decision (e.g., [55]).

Common to all these approaches is the basic assumption that the training data is representative of the underlying data distribution and properly pre-processed accordingly.

d) Bias and Fairness in Machine Learning: Within the machine learning community, there is a lively discussion on how to recognize and deal with biased data. In her popular science book on “Weapons of math destruction” [56], the author Cathy O’Neill shows how algorithms can be unfair, even if they perfectly reflect the data: If the data itself is biased, the trained machine learning algorithm will reflect this bias and, accordingly, make unfair decisions.

There is a class of learning algorithms that are subgroup-fair [57]. Here, a subgroup is defined as a group of instances that have the same attribute values, such as all males, or all females above the age of 50 who own a car. If we want those subgroups to have equal opportunity, then we need to make sure that a) the classifier is as accurate on the subgroup as it is on the whole population (e.g., equal error rate), and b) if it makes errors, it makes the same errors in the subgroup as in the whole population (e.g., equal false positive rates) [58].

While research on algorithms fairness is still quite young, it turns out fairness is a concept that is difficult to encode algorithmically: [59] identified 21 different definitions that have been used in literature. An additional challenge is to assess the impact of today's fair decisions – we expect them to improve the situation for discriminated subgroups – but this is not necessarily the case for different fairness criteria [59].

VI. Discussion and Conclusion

We presented a small example to illustrate the basic approach of data preparation and model learning in a distributed system, and derived challenges that arise in distributed machine learning. In real-world settings the situation can often be much more complex. In huge, unstructured data sets, data distribution might not be that obvious as in our small-world example with tabular data. There may be many missing values and different data sets might be pre-processed differently. Veracity and velocity of the underlying data may have a non-negligible impact on trustworthiness of machine learning. More complex (distributed) ML algorithms do not even allow for as straightforward explanations as is possible for our small decision trees. Moreover, several of our presented challenges might be in conflict and not all requirements might be fulfilled at the same time in real-world applications. There might for example be a trade-off between fairness of machine learning and privacy requirements.

While our example is very simplified and any classification model would have problems to construct a reliable prediction based on such a small training data set, it nevertheless illustrates the most important points:

- Without knowledge about the model construction workflow and settings (such as model hyperparameters, decision aggregation function in ensemble methods), it is impossible to reliably trace decisions back.
- Without knowledge about data provenance (including bias in the data set and any data imputations applied) we are unable to assess the trustworthiness of the resulting machine learning decision.

Thus, by considering data provenance, models can gain in accuracy and fairness, and explanations can become more holistic and truthful.

We propose to systematically exploit data provenance, gathered throughout the data preparation pipeline, to improve the outcome of the machine learning pipeline. In particular, we present our vision of deriving end-to-end explanations of data analyses. We even go so far as to claim that there can be no truthful and complete explainability in machine learning without taking data provenance into account.

The authors in [60] make an even more radical demand and request that the recording of data provenance must start even before the raw data is collected. They argue that data is always collected for specific purpose, and that this purpose must be well-documented. Knowing why the raw data was collected in the first place, we could tell a more truthful story as to how the machine learning model came to be. We regard this as a very promising, long-term direction for future work that requires interdisciplinary discussions between the producers of the raw data, the data engineers and the machine learning experts.

References

[1] S. Ghosh, *Distributed Systems: An Algorithmic Approach, Second Edition*, 2nd ed. Chapman & Hall/CRC, 2014.

[2] N. Diakopoulos, "Accountability in algorithmic decision making," *Commun. ACM*, vol. 59, no. 2, pp. 56–62, Jan. 2016.

[3] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"," *ArXiv e-prints*, Jun. 2016.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[6] L. Liu and M. T. Özsu, *Encyclopedia of Database Systems*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[7] N. Papernot and P. D. McDaniel, "Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning," *CoRR*, vol. abs/1803.04765, 2018.

[8] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, "Generating Counterfactual Explanations with Natural Language," in *Proc. WHI*, Jun. 2018. [Online]. Available: https://arxiv.org/abs/1806.09809

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proc. KDD 2016*, 2016, pp. 1135–1144.

[11] S. Schelter, J.-H. Böse, J. Kirschnick, T. Klein, and S. Seufert, "Automatically tracking metadata and provenance of machine learning experiments," in *Machine Learning Systems Workshop at NIPS*, 2017.

[12] M. Herschel, R. Diestelkämper, and H. Ben Lahmar, "A survey on provenance: What for? What form? What from?" *VLDB J.*, vol. 26, no. 6, pp. 881–906, 2017.

[13] P. Buneman and W.-C. Tan, "Data provenance: What next?" *ACM SIGMOD Record*, vol. 47, no. 3, pp. 5–16, 1 2019.

[14] Y. Cui and J. Widom, "Lineage tracing for general data warehouse transformations," *The VLDB Journal*, vol. 12, no. 1, pp. 41–58, May 2003.

[15] S. Abiteboul, M. Arenas, P. Barceló, M. Bienvenu *et al.*, "Research Directions for Principles of Data Management (Abridged)," *SIGMOD Rec.*, vol. 45, no. 4, May 2017.

[16] J. Singh, J. Cobbe, and C. Norval, "Decision provenance: Capturing data flow for accountable systems," *CoRR*, vol. abs/1804.05741, 2018. [Online]. Available: http://arxiv.org/abs/1804.05741

[17] S. Lee, S. Köhler, B. Ludäscher, and B. Glavic, "A SQL-Middleware Unifying Why and Why-Not Provenance for First-Order Queries," in *Proc. ICDE 2017*, 2017, pp. 485–496.

[18] D. Deutch, Y. Moskovitch, I. Polak, and N. Rinetzky, "Towards Hypothetical Reasoning Using Distributed Provenance." in *Proc. EDBT 2018*, 2018, pp. 461–464.

[19] T. Müller, B. Dietrich, and T. Grust, "You Say 'What', I Hear 'Where' and 'Why'? (Mis-)Interpreting SQL to Derive Fine-Grained Provenance," *PVLDB*, vol. 11, no. 11, pp. 1536–1549, 2018.

[20] P. Buneman, S. Khanna, and W. C. Tan, "Why and Where: A Characterization of Data Provenance," in *Proc. ICDT 2001*, 2001, pp. 316–330.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc., February 2008, no. 978-0387310732.

[22] D. J. Stekhoven and P. Buehlmann, "MissForest – non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.

[23] A. Doan, A. Halevy, and Z. Ives, *Principles of Data Integration*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[24] D. Suciu, D. Olteanu, R. Christopher, and C. Koch, *Probabilistic Databases*, 1st ed. Morgan & Claypool Publishers, 2011.

[25] J. Jestes, K. Yi, and F. Li, "Building wavelet histograms on large data in mapreduce," *PVLDB*, vol. 5, no. 2, pp. 109–120, 2011.

[26] B. Yildiz, T. Büyüktanir, and F. Emekçi, "Equi-depth histogram construction for big data with quality guarantees," *CoRR*, vol. abs/1606.05633, 2016.

[27] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: A Warehousing Solution over a Map-Reduce Framework," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1626–1629, Aug. 2009.

[28] B. Ganesh Babu, S. T P, and S. Kumar, "Dynamic colocation algorithm for hadoop," 09 2014, pp. 2643–2647.

[29] C.-W. Lee, K.-Y. Hsieh, S.-Y. Hsieh, and H.-C. Hsiao, "A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments," *Big Data Res.*, vol. 1, no. C, pp. 14–22, Aug. 2014. [Online]. Available: https://doi.org/10.1016/j.bdr.2014.07.002

[30] V. Ubarhande, A. Popescu, and H. Gonzlez-Vlez, "Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments," in *2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*, July 2015, pp. 217–224.

[31] A. Al-Ghezi and L. Wiese, "Adaptive workload-based partitioning and replication for rdf graphs," in *International Conference on Database and Expert Systems Applications*. Springer, 2018, pp. 250–258.

[32] L. Wiese, T. Waage, and F. Bollwein, "A replication scheme for multiple fragmentations with overlapping fragments," *The Computer Journal*, vol. 60, no. 3, pp. 308–328, 2017.

[33] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[34] E. H. Simpson, "The interpretation of interaction in contingency tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 13, no. 2, pp. 238–241, 1951. [Online]. Available: http://www.jstor.org/stable/2984065

[35] J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," *Science Advances*, vol. 4, no. 1, 2018. [Online]. Available: http://advances.sciencemag.org/content/4/1/eaao5580

[36] T. Munzner, *Visualization Analysis and Design*, ser. AK Peters Visualization Series. CRC Press, 2014.

[37] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings of the IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

[38] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: ACM, 2018, pp. 582:1–582:18. [Online]. Available: http://doi.acm.org/10.1145/3173574.3174156

[39] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Transactions on Visualization and Computer Graphics*, 2018. [Online]. Available: https://fredhohman.com/visual-analytics-in-deep-learning/

[40] L. Wiese, *Advanced Data Management: For SQL, NoSQL, Cloud and Distributed Databases*. Walter de Gruyter GmbH & Co KG, 2015.

[41] J. Salas and V. Torra, "A general algorithm for k-anonymity on dynamic databases," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2018, pp. 407–414.

[42] F. Bollwein and L. Wiese, "Keeping secrets by separation of duties while minimizing the amount of cloud servers," *T. Large-Scale Data- and Knowledge-Centered Systems*, vol. 37, pp. 1–40, 2018.

[43] F. Provost and V. Kolluri, "A survey of methods for scaling up inductive algorithms," *Data Min. Knowl. Discov.*, vol. 3, no. 2, pp. 131–169, Jun. 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1009876119989

[44] G. Karvounarakis, Z. G. Ives, and V. Tannen, "Querying Data Provenance," in *Proc. SIGMOD 2010*, 2010, pp. 951–962.

[45] H. Park, R. Ikeda, and J. Widom, "Ramp: A system for capturing and tracing provenance in mapreduce workflows," in *37th International Conference on Very Large Data Bases (VLDB)*. Stanford InfoLab, August 2011. [Online]. Available: http://ilpubs.stanford.edu:8090/995/

[46] M. Interlandi, A. Ekmekji, K. Shah, M. A. Gulzar, S. D. Tetali, M. Kim, T. Millstein, and T. Condie, "Adding data provenance support to apache spark," *The VLDB Journal*, vol. 27, no. 5, pp. 595–615, Oct. 2018. [Online]. Available: https://doi.org/10.1007/s00778-017-0474-5

[47] B. Saha, "Data science & engineering platform: Data lineage and provenance for apache spark," 2018, online https://de.hortonworks.com/blog/data-science-engineering-platform-data-lineage-provenance-apache-spark/; published 11-Dec-2018.

[48] A. Chen, Y. Wu, A. Haeberlen, B. T. Loo, and W. Zhou, "Data Provenance at Internet Scale: Architecture, Experiences, and the Road Ahead," in *Proc. CIDR'17*, 2017.

[49] D. Szafron, P. Lu, R. Greiner, D. Wishart, Z. Lu, B. Poulin, J. Anvik, and C. Macdonell, "Proteome Analyst – transparent high-throughput protein annotation: function, localization and custom predictors," in *Proc. ICML Workshop on Machine Learning in Bioinformatics*, 2003.

[50] R. Tomsett, D. Braines, D. Harborne, A. Preece, and S. Chakraborty, "Interpretable to Whom? A Role-based Model for Analyzing Interpretable Machine Learning Systems," in *Proc. WHI*, Jun. 2018.

[51] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 93:1–93:42, Aug. 2018.

[52] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[53] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning," *ArXiv e-prints*, May 2018.

[54] C. Seifert, A. Aamir, A. Balagopalan, D. Jain *et al.*, "Visualizations of Deep Neural Networks in Computer Vision: A Survey," in *Transparent Data Mining for Big and Small Data*, T. Cerquitelli, D. Quercia, and F. Pasquale, Eds. Springer, 2017, pp. 123–144.

[55] K. Leino, L. Li, S. Sen, A. Datta, and M. Fredrikson, "Influence-Directed Explanations for Deep Convolutional Networks," *CoRR*, vol. abs/1802.03788, 2018.

[56] C. O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. New York, NY, USA: Crown Publishing Group, 2016.

[57] M. J. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness," in *Proc. ICML*, vol. abs/1711.05144, 2018. [Online]. Available: http://arxiv.org/abs/1711.05144

[58] Z. Zhang and D. Neill, "Identifying Significant Predictive Bias in Classifiers," in *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, ser. FATML, 2017.

[59] L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt, "Delayed impact of fair machine learning," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 80. JMLR.org, 2018, pp. 3156–3164.

[60] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan *et al.*, "Datasheets for Datasets," *CoRR*, vol. abs/1803.09010, 2018.