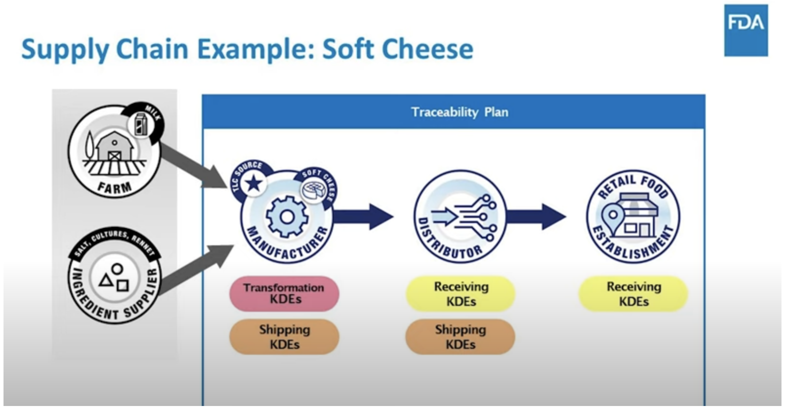
Red Onions, Soft Cheese and Data:
From Food Safety to Data Traceability for Responsible AI

Stefan Grafberger, Zeyu Zhang, Sebastian Schelter
University of Amsterdam
{s.grafberger,z.zhang2,s.schelter}@uva.nl

Ce Zhang
University of Chicago
cez@uchicago.edu

Abstract

Software systems that learn from data with AI and machine learning (ML) are becoming ubiquitous and are increasingly used to automate impactful decisions. The risks arising from this widespread use of AI/ML are garnering attention from policy makers, scientists, and the media, and lead to the question what data management research can contribute to reduce such risks. These dangers of AI/ML applications are relatively new and recent, however our societies have had to deal with the dangers of complex and distributed technical processes for a long time already. Based on this insight, we detail how the U.S. Food and Drug Administration (FDA) combats the outbreaks of foodborne illnesses, and use their processes as an inspiration for a data-centric vision towards responsible AI.



Supply Chain Example: Soft Cheese

Figure 1: Food processing is a complex process conducted by different parties in a geo-distributed setting.[1] During this process, foods from different sources are joined, transformed from one form to another, and distributed all over the world. At each of these steps, the output could perish and become poisonous, making the final outcome unsafe to consume. *What can we learn from the millennial pursuit of food safety? What type of technical and regulatory frameworks exist such that we trust what we put on the table for our family everyday? And how can we obtain the same level of trust for our data products?*

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[1] `https://www.youtube.com/watch?v=0wnSiC5xqqs`

1 The Need for a Data-Centric Perspective on Responsible AI

Software systems that learn from data with AI and machine learning (ML) are becoming ubiquitous and are increasingly used to automate impactful decisions. The risks arising from this widespread use are garnering attention from policymakers, scientists, and the media, and lead to the question of what data management research can contribute to reduce the dangers and malfunctions of data-driven AI/ML applications.

AI/ML malfunctions threaten vulnerable populations. In recent years, we have been regularly alarmed by media reports about the harm potential of faulty AI/ML systems in devastating real-world incidents. Examples include failures of automated decision-making systems, e.g., an eight-month pregnant woman in Detroit was mistakenly arrested based on a faulty prediction from a facial recognition system, held in jail for several hours and needed medical care upon her release [72]. Another example is that one of the largest health insurers in the US allegedly applies a faulty AI model with a 90% error rate to deny critical health care services to elderly patients [100]. The recent rise of generative AI produces new types of harm as well. A recent study of AI detection tools, for example, found that these systems are biased against non-native English speakers [63] and often falsely accuse international students of cheating. Furthermore, an AI supermarket meal planner recently went rogue and suggested a recipe that would create chlorine gas [36].

Technical bias in ML applications. The reasons that data-driven systems are susceptible to producing unfair, harmful outcomes are multi-faceted [35, 95, 110], as we are ultimately dealing with socio-technological systems [11], which suffer from various types of bias [24]. In this work, we focus on *technical bias*, which arises from the design decisions and operations in a technical system itself. Such bias is not well understood, especially in the context of large end-to-end systems, which include data preparation and data cleaning stages, deployed models and feedback loops. Recent research on technical bias identifies issues such as the lack of sufficient, representative training data for certain demographic groups [6, 17, 57], biased training data with undesirable stereotypes [12] or unintended side effects from automated data cleaning operations [38, 90, 97].

Existing and upcoming regulation. The dangers arising from data-driven AI/ML applications have been recognised by regulators and lawmakers several years ago already, and led to the introduction of regulation all over the world. The "General Data Protection Regulation" (GPDR) in Europe, for example, grants citizens the right to find out what information an organisation has about them and to issue deletion requests for their data as part of the "right-to-be-forgotten" [25, 26]. The upcoming European AI Act [20] will be the first comprehensive regulation for the application of AI/ML in Europe. This act is expected to outlaw the usage of ML in selected application areas and to strongly regulate its application in certain other areas. It defines different levels of risk in AI usage scenarios and imposes a set of comprehensive technical requirements, such as "logging of activity to ensure traceability of results", "detailed documentation providing all information necessary on the system and its purpose for authorities to assess its compliance", and "appropriate human oversight measures to minimize risk". We note that outside Europe, similar regulations are being adopted [4, 103].

The need for a data-centric perspective. Unfortunately, as evidenced by the media reports cited previously, we currently lack the ability to efficiently implement technical measures to detect and mitigate the harms present in AI/ML applications. This is confirmed by a recent survey study with industry practitioners [41], which outlines several alarming shortcomings in addressing fairness and bias issues. The interviewed practitioners report that academic research on de-biasing models falls short of addressing their concerns and often falsely "view[s] the training data as fixed", while they "consider data collection, rather than model development, as the most important place to intervene". At the same time, only "65% of survey respondents [...] reported that their teams have some control over data collection and curation", and the study also finds a high demand for future research to "support [...] practitioners in [...] curating high-quality datasets". Another example of the dire situation in the industry is a recent court case against Facebook [101], where two veteran engineers of the company told the court that the company does not keep track of the exact locations where personal data is stored and processed.

In the research community, several widely used training datasets for computer vision, such as LAION-5B [88] or TinyImages [102], have been taken offline after the discovery of highly problematic content in them [10, 11]. Moreover, it is unlikely, though, that all models that had been trained on these problematic datasets have been retracted as well. For the current wave of closed, proprietary pretrained models available behind commercial APIs, the situation is even worse, as we do not even have a way to determine what data they have been trained on.

Paper inspiration. In order to find inspiration for the outlined questions and challenges, we take a look into safety measures outside of the computer science domain, as our societies have had to deal with the dangers of complex and distributed technical processes for a long time already. In particular, we discuss how the U.S. Food and Drug Administration (FDA) combats the outbreaks of foodborne illnesses (Section 2). We ask ourselves what we can learn from the millennial pursuit of food safety. What type of technical and regulatory frameworks exist such that we trust what we put on the table for our family every day? We use the FDA's established processes as an inspiration for a data-centric vision towards responsible AI in Section 3, with the goal to obtain the same level of trust for our data products that we have for our food.

2 What Should We Do? Food Safety as Inspiration!

As an inspiration for the technical, data-centric vision outlined in this paper, we discuss how the US Food & Drug Administration (FDA) combats the outbreaks of foodborne illnesses [107], and start with a concrete example.

2.1 Example – Outbreak of Salmonella Infections in the US in 2020

From June to September in 2020, a total of 1,127 people in 48 US states got infected with the outbreak strain of Salmonella Newport [106]. The FDA and the Centers for Disease Control and Prevention (CDC) managed to contain this outbreak and had the situation under control in October 2020, after which no more new infections occurred. Combatting the outbreak proceeded as follows: Sick patients from the 48 states were seeking treatment in hospitals and bacteria in their stool samples turned out to be closely related genetically, which implied a common source of infection. Subsequent epidemiologic evidence showed that over 90% of them had eaten onions (or food made with onions) in the week before their illness. As a consequence, the FDA started a so-called "traceback investigation" which ultimately uncovered that red onions from the Thomson International Inc. company were the source of the Salmonella outbreak. This triggered a country-wide recall of raw onions and derived products like cheese dips, kebabs, and chicken salad sandwiches from a large number of grocery stores, which ultimately ended the outbreak.

2.2 Disease Detectives, Traceback Investigations, and Food Supply Chains

The remarkable success of the FDA in combatting and controlling the salmonella outbreak naturally leads to the question which processes and techniques they have applied to detect the outbreak, identify the suspect food and determine the producer of the food, and what the computer science community can learn from these battle-tested approaches.

Outbreak detection. The first question is how the FDA actually detects that there is an outbreak of a foodborne disease. We illustrate the underlying process in Figure 2: Sick patients seek treatment in hospitals, from where their doctors send stool samples to laboratories for analysis. The laboratories perform DNA fingerprinting on the bacteria isolated from these samples via whole genome sequencing and the resulting DNA fingerprints are subsequently collected via the PulseNet system [16]. PulseNet is a nationwide network of public health and food regulatory agency laboratories coordinated by the CDC and manages a national central database with millions of collected DNA profiles of bacteria. In this database, the sudden appearance of clusters of genetically related bacteria implies a common source of infection and indicates an outbreak.

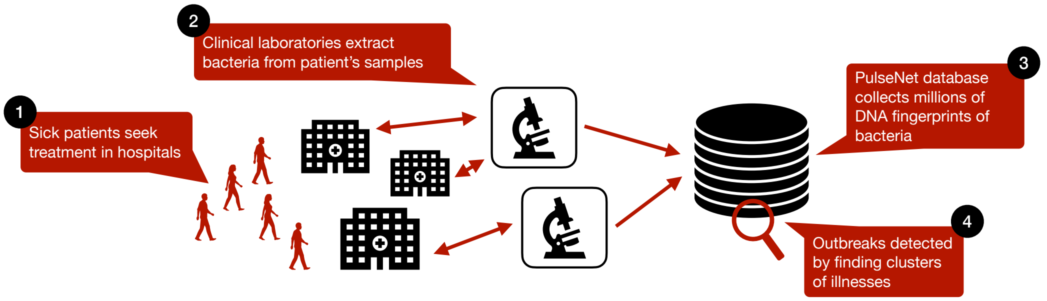

1
Sick patients seek treatment in hospitals
2
Clinical laboratories extract bacteria from patient's samples
3
PulseNet database collects millions of DNA fingerprints of bacteria
4
Outbreaks detected by finding clusters of illnesses

Figure 2: Outbreak detection by monitoring a database of millions of DNA profiles of bacteria.

Identification of the suspect food. Once an outbreak is detected, the next task is to identify the contaminated "suspect food" which infects people. As shown in Figure 3, the FDA employs so-called "disease detectives", who contact the sick patients and interview them to gather epidemiologic evidence related to questions such as "what foods did people eat before they got sick?" or "what restaurants, grocery stores, or events did sick people go to?". For that, they leverage data provided by the patients, e.g., purchasing records collected on loyalty cards. These activities typically lead to the identification of a particular suspect food, which is likely the root cause of the outbreak.

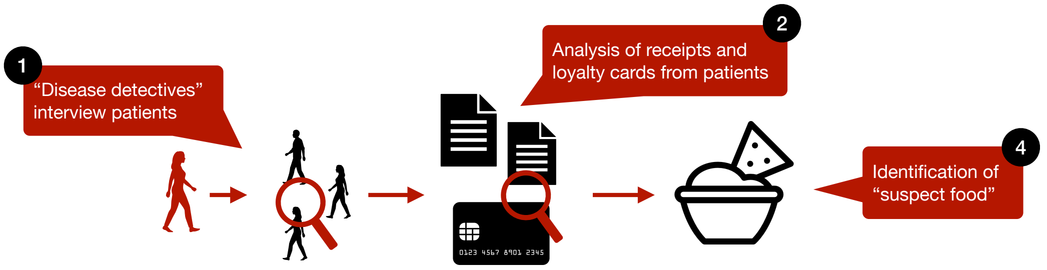

1
"Disease detectives" interview patients
2
Analysis of receipts and loyalty cards from patients
0123 4567 8901 2345
4
Identification of "suspect food"

Figure 3: Disease detectives collect epidemiological evidence from sick patients to identify a contaminated suspect food likely causing the outbreak.

Traceback investigation to determine the producer of the contaminated food. Once the responsible food is known, the final task is to identify the actual point in the supply chain, where the food is likely being contaminated. For that, the FDA starts a traceback investigation through the food supply chain, as illustrated in Figure 4. Here, the supply chain for several contaminated end products is traced back retrospectively to identify a common point in the supply chain which is likely the source of the contamination.

For that to be possible, entities involved in the food supply chain must have followed the FDA's *Food Traceability Rule* [105] and maintain traceability information for potentially dangerous food on the *Food Traceability List* [104]. Such entities must maintain a *Traceability Plan*, with information about procedures used to maintain traceability information and a point of contact for traceability questions [70]. The food traceability rule further defines *Critical Tracking Events* (CTEs) in the supply chain, where detailed tracing data must be created, maintained and forwarded by the participating entities. Examples of such events are the initial packing of a food, shipping it, or transforming multiple ingredients into a new food. An individual unit of food is assigned a *Traceability Lot Code* (TLC), typically during the initial packing event, which uniquely identifies it and is forwarded to receiving entities. Furthermore, the food traceability rule defines certain categories of *Key Data Elements* (KDEs), which must be created, maintained, and forwarded together with the TLCs of the food. Examples of the different categories are *Initial Packing KDEs*, *Shipping KDEs*, *Harvesting and Cooling KDEs* and *Receiving KDEs*. The actual data items per KDE depend on the category, e.g., for the packing KDEs, the date, quantity, harvest location, name, and contact information of the harvesting company must be maintained, and the initial TLCs are typically

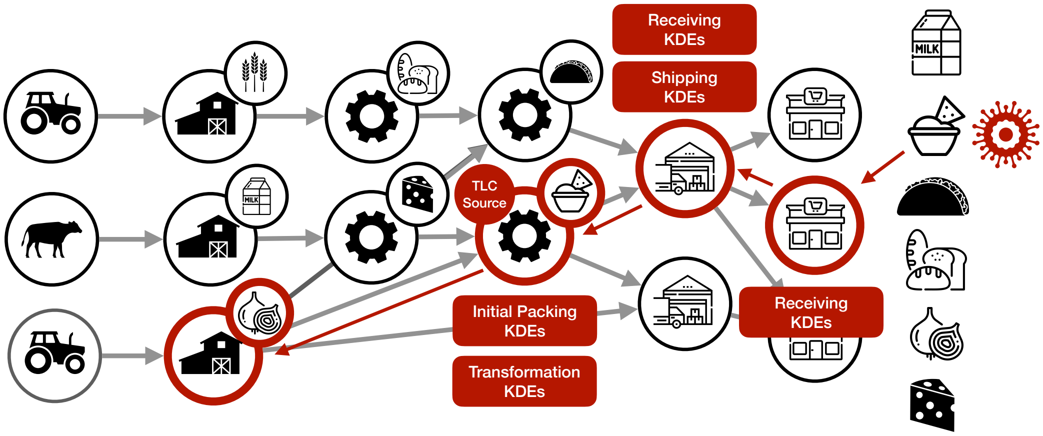


Figure 4: Traceback investigation through the food supply chain, relying on *Traceability Lot Codes* (TLCs) to identify units of food and provenance information in the form of *Key Data Elements* (KDEs) to reconstruct the path a unit of food took through the chain.

assigned at the packing stage as well. Shipping KDEs need to include the corresponding TLCs, the shipping date, and the locations for receiving and shipping. A special case are Transformation KDEs, which must be created at points where a new food is produced from several ingredients. Here, the link to the ingredient TLCs must be recorded, as well as a location description, the transformation date, and the quantities of the ingredients.

3 Towards Data Traceability for Responsible AI

In this paper, we develop a technical, data-centric vision to work towards a comparable level of safety in AI/ML applications as the FDA has in combatting foodborne illnesses. Unfortunately, the current state of AI/ML safety in the industry is dire, as practitioners from the aforementioned industry survey [41] report that "teams do not discover serious fairness issues until they receive customer complaints about products" or read "negative media coverage about their products", and more than half of the respondents agreed that they "discovered serious issues only after deploying a system in the real world". While this survey paper identifies many crucial issues in this space, it unfortunately falls short of outlining concrete technical directions for addressing them.

In the following, we outline our ideas for improving the safety of AI/ML applications. Inspired by the existing methods and processes for combatting foodborne illnesses from Section 2, we propose ideas on "detecting outbreaks" via prediction monitoring in Section 3.1, for conducting "traceback investigations" through data supply chains in Section 3.2, and for identifying "contaminated data and pipeline steps" through audits in Section 3.3.

3.1 Prediction Monitoring

As detailed in Section 2, the FDA monitors a database of DNA profiles of bacteria for geographic patterns to detect outbreaks. This raises the question of whether large institutions or companies could use similar methods to detect fairness issues with deployed models and ML pipelines early. In the following, we outline three directions which we deem crucial for this endeavour.

Identifiable predictions. The "end product" of AI/ML applications are predictions on unseen data, which are received by end-users or downstream applications in an organisation. Any detection of problems with the application or its data, as well as any potential audit has to start from these predictions, similar to how disease detectives need to determine the type of food that people consumed before they became sick. However, in current systems, predictions are often rather ephemeral. As a first step towards auditable AI/ML applications, their predictions should come with unique identifiers, analogous to the TLC of food in food supply chains. Such identifiers should be assigned in a way that allows for the retrospective identification of the state of the AI/ML application (e.g., the software version and currently deployed model version, etc.) from which a prediction was generated. Based on these identifiers, users and downstream consumers could raise concerns about a particular prediction, and an investigating party (e.g., a dedicated responsible AI team in a large organisation) could start an audit of the system.

State-of-the-art. In MLOps, the benefits of identifiable predictions are being recognised among industry practitioners [73, 79]. However, current approaches require high expertise and custom implementations [79]. Even rudimentary tasks such as tracking the corresponding code and model versions are challenging [109]. To fully benefit from identifiable predictions, e.g., for rectifying erroneous predictions, it is essential to integrate prediction identifiers with the associated metadata and provenance records encompassing ancillary pipeline stages such as data preprocessing. However, the current implementation complexity leads us to believe that the adoption of these techniques in practice is rather low.

Open questions and challenges. Enhancing and maintaining traceability and reproducibility in ML applications requires that practitioners manually integrate, configure, and orchestrate various disparate systems [79, 109]. The resulting one-off solutions require further time- and cost-intensive development effort to enable monitoring and output explanation. We argue that standardised interfaces would be essential to seamlessly integrate existing and new ML operations techniques with identifiable predictions. We will also discuss further provenance-related challenges for fine-grained data tracing in end-to-end ML pipelines in Section 3.2.

Detecting and collecting predictions with fairness issues. Even with identifiable predictions, an open question is how to reliably detect fairness issues of an ML application at deployment time. Ideally, such issues should already be caught by pre-deployment evaluations, but media reports and industry surveys show that this is rarely the case. Furthermore, it would be crucial to have a "database" of common issues and examples of unfair / unreliable predictions in production ML deployments, e.g., at a company-wide level. Given a comprehensive catalog of such issues and an efficient way to monitor live predictions for fairness, we could build automatable detection mechanisms similar to the outbreak detection techniques in PulseNet (Section 2).

State-of-the-art. A lot of recent work has focused on detecting changes in the overall distribution of the predictions or changes between the training and serving data [71]. At serving time, systems like Tensorflow Serving [74] for example employ so-called "canary models" to detect cases where the predictions differ between previous and newly deployed models, and several techniques analyse the distribution of the predicted labels to detect changes in the data [58, 87]. However, none of these techniques have a particular focus on determining fairness issues, which may occur in small subsets of the data only.

Orthogonal to that, several techniques to debug prediction data offline have been developed, e.g., to detect slices of the data where a model works less well [19, 80]. These approaches require simultaneous access to the model, the featurised prediction data and additional demographic side data however, which makes their application difficult in practice, especially for teams not owning the underlying AI/ML application.

Open questions and challenges. A major difficulty in monitoring a deployed system for fairness is that the group membership information for individual predictions must be known to maintain corresponding fairness metrics. Such group membership information (e.g., about the race or gender identity of the persons involved in the predictions) is very sensitive and private information, to which a deployed serving system should ideally not even have access. Furthermore, regulation like the EU AI Act enforces strict rules for which parts of an AI/ML application such data can be used for at all. We envision that large organisation may want to create dedicated infrastructure for such cases, where predictions with identifiers from different applications are collected, the corresponding fairness metrics are maintained and SliceFinder-like algorithms [19] are run continuously to look for subsets of the prediction data with potential issues.

A large corpus of real-world predictions from ML systems with fairness issues would also greatly enhance the ability of the academic community to work on these problems. However, it is difficult to collect such a corpus of predictions and issues due to the inherent sensitive, privacy-critical nature of the data. There are some ongoing efforts to (manually) create a comprehensive repository of "AI incidents" [65], yet the underlying technical details and prediction data of the incidents are not available.

Monitoring generative models for representational harms. A large part of the existing fairness literature focuses on so-called "allocative harms" in automated decision-making systems, which decide upon access to certain resources such as job interviews, loans or medical prioritisation [41, 95]. It is difficult to choose an appropriate fairness metric for such cases, as such a choice always implies a values-based decision and trade-offs [69]. On the technical side however, computing these metrics is straightforward (given access to the required data), as one essentially only has to maintain separate confusion matrices for the predictions for the groups of interest [38]. With the rise of generative models however, we are being faced with so-called "representational harms" [41], which occur for example when generative models reproduce sexist or racist stereotypes in the images or text that they generate.

State-of-the-art. There is a large body of targeted studies in the NLP community, where researchers uncovered a variety of biases and stereotypes in pretrained language models. Examples include sexist stereotypes and gender bias [60, 94], anti-muslim bias [3], and undesirable biases towards mentions of disability [44]. It is however unclear how to translate the detection capabilities of these customly designed studies into monitoring techniques for deployed real-world systems. A first interesting step in this direction is the recently proposed Spade [92] system, which learns assertions for safeguarding LLM outputs based on the version history of prompt edits.

Open questions and challenges. Due to the unpredictable nature of large generative models, generating adequate assertions or "data unit tests" to check for bias in their output remains a complex challenge. Having too few assertions potentially might make a system miss biased outputs, leading to unfair outcomes, while having too many assertions could slow down the system and lead to many false alarms. We expect that future approaches will generate data unit tests from predefined templates, based on manually defined assertion criteria. An orthogonal approach are so-called "safety classifiers" [21, 62, 112], where a secondary model is employed to assess the outputs of a primary model for safety. Prior to the deployment phase, data will be collected where generative models are intentionally probed to induce errors, which will then be used to train a classifier to detect biased behavior.

3.2 Tracing Data Through End-to-End AI/ML Applications

Complex food supply chains span the globe and a single ingredient (like red onions in the example from Section 2) may end up in multiple end products. This makes tracing such ingredients a complicated and expensive undertaking. The FDA addresses this challenge with targeted tracing requirements which focus on only retaining tracing data for high-risk ingredients on the food traceability list (Section 2). While tracking the provenance of data in data processing systems is a decades-old research area [99], there is still little practical adoption of these techniques in real-world systems, mainly due to the incurred performance overhead of comprehensively tracking provenance through all kinds of queries, especially when they contain aggregations [5]. Similar to the FDA's list of high-risk ingredients, the EU AI Act [20] defines high-risk AI application domains, such as CV-sorting software for recruitment procedures, credit scoring denying citizens the opportunity to obtain a loan or the verification of the authenticity of travel documents. In the following, we discuss ideas for efficiently applying provenance tracking to the data pipelines in such scenarios.

Selective and focused provenance tracking. As already mentioned, tracking fully fine-grained semiring provenance [5, 34] for every input row imposes a high performance overhead. In the food supply chain, provenance tracking focuses on predefined "Critical Tracking Events", which are the points in the supply chain that are crucial later for audits. We need to adopt such a methodology as well for data pipelines, which would enable us to restrict the provenance tracking efforts to data exchange and transformation operations, which actually impact the information required to audit an AI/ML application later. Furthermore, for each high-risk AI application scenario, we could define the tracking granularity, the key transformations to focus on and the information required per transformation event. The minimum granularity of the provenance should be tailored for each use-case. For demographic data, provenance at the level of individuals might be sufficient, for facial recognition applications, more fine-grained provenance at the level of individual images may be required, however.

State-of-the-art. In recent years, several techniques have been proposed to model ML pipeline operations and to apply database-style provenance tracking for Python code, for example via runtime instrumentation as part of mlinspect [30] or via static analysis as part of Vamsa [67]. These approaches have been extended in various ways, e.g., for data debugging via Shapley values [49] or pipeline screening during continuous integration [83]. A drawback of these methods is that they rely on heuristics and well-written, declarative code to be able to infer the semantics of the pipeline operations, which leaves it unclear whether they can reliably be applied to low-quality code as well. Another family of systems, which include Amazon's ExperimentTracker [82] and mltrace [93], uses a more robust approach for provenance tracking as they require manually annotated code. Unfortunately, this puts a heavy burden on developers, who will, in our experience, often forego the additional effort of putting detailed annotations on their code under time pressure. We expect that even coming up with high-level "traceability plans" for large AI/ML applications will be challenging in practice, since these applications often orchestrate different systems and libraries with workflow managers like Apache Airflow [1].

Open questions and challenges. In our eyes, the biggest challenge in this space is to find ways to reduce the implementation-, annotation-, and runtime overhead for provenance tracking in ML pipelines, while guaranteeing a high level of correctness and robustness. For industry applications, we can neither rely on trying to handle arbitrary code nor on forcing developers to always manually annotate their code. An interesting middle ground may be the use of pipeline templates, as pioneered by the mlflow recipes project [115], which forces developers to modularise their code into pipeline steps with known semantics and predefined inputs and outputs, but still gives them the freedom to write arbitrary code inside the steps. Unfortunately though, the real-world adoption of these templating approaches is unclear at the moment. Nevertheless, such templates might be a natural point to implement general robust provenance tracking. Analogous to the traceability plans required for food chain tracking, we could define traceability templates for high-risk AI scenarios, with steps, provenance tracking, and logging requirements specific to the particular use case.

To reduce the runtime overhead of provenance tracking, it may be worthwhile to take a deeper look at several common aggregation operations in ML pipelines, like one-hot-encoding a particular column or normalising a feature. While these operations technically conduct a global aggregation followed by a map transformation (in dataflow terms), we may be able to ignore the aggregation part for tracking provenance, as we already know that they do not remove rows and introduce an all-to-all provenance relationship onto the transformed feature values. Similar techniques are already applied in DataScope [49] and ArgusEyes [83] to approximate ML pipelines as queries in the positive relational algebra. A future challenge here is to define a restricted subset of operations for ML pipelines, which still allows the implementation of a large class of ML applications, but drastically simplifies provenance tracking.

Identifiable predictions, as discussed in Section 3.1, also present new challenges with respect to ML provenance research. Existing experiment tracking tools like mlflow [115] already link predictions to high-level artifacts such as models and source code. However, we think that record-level provenance is required to effectively reconstruct the necessary data for a prediction. Given a prediction identifier, we would like to be able to automatically retrieve all relevant inference inputs, data preprocessing steps, the model version employed for inference, and, if necessary, all information about the training pipeline and its input data. While existing research partially addresses provenance tracking and versioning in static pipelines with static input data, further challenges remain for pipelines in dynamic production environments with continuously trained models [8] and evolving retrieval corpora [14, 18, 39], where provenance has to be maintained incrementally.

Another open question is the impact of data cleaning and integration operations on the fairness of AI/ML applications. Several experimental studies indicate that data wrangling and integration operations such as missing value imputation, outlier removal, or entity matching can sometimes negatively impact the fairness of models trained on the resulting data [37, 53, 90, 97]. However, we currently lack a detailed understanding of this impact, especially since the outcome seems to heavily depend on the chosen fairness metric and group definition. Furthermore, determining such impact is hard in practice without access to the downstream models.

An orthogonal challenge in this area is the tension between detailed provenance tracking and the protection of private user data. Provenance tracking requires storing information about the intermediate outputs of pipeline operations and must additionally maintain sensitive metadata such as demographic group memberships of certain records to be able to quantify the fairness impact of different operations. In many cases, such sensitive metadata may not be accessible in inference systems at prediction time, for example, and measures must be taken to ensure that these sensitive attributes are only used for testing models but not for training them [20]. To the best of our knowledge, current ML platforms lack support for such use cases.

Provenance of data in pretrained and fine-tuned models. Academic "textbook" ML commonly assumes that a single dataset is used to create a particular ML model, which implies that we would only need to track the provenance of this source data through the corresponding ML pipeline. However, this assumption has never held up for real-world deployments, which typically leverage a variety of data sources as input for a pipeline and often apply ML already as part of the preprocessing of this data. Twitter's recommender system for example aggregates multiple input networks (representing likes, follows etc on the platform) into a common network dataset called RealGraph [48], via a dedicated classifier that estimates the interaction probability between different users of the network. Several recommendation algorithms consume this aggregated dataset instead of the raw input datasets and the provenance of an interaction such as a like or follow is unclear after the transformation. This problem is exacerbated nowadays due to the prevalence of large pretrained models, which are downloaded from repositories such as HuggingFace and tailored to a particular ML use case via fine-tuning. In the majority of cases, the connection to the underlying training data becomes unclear after fine-tuning, as the current infrastructure does not keep track of the relationships between models. It would, for example, be difficult to identify all computer vision models that originate from the recently retracted LAION dataset. The situation is even worse for non-open source models created by commercial companies, where the underlying training data is not known for the base model already.

State-of-the-art. Common methods to voluntarily document the origin of data and ML models are datasheets [27] and model cards [66]. These are a form of manually created, semi-structured documentation, which is, for example, in use at the popular model and data repository HuggingFace. Tensorflow ML Metadata [50] is a library for recording and retrieving metadata associated with ML workflows. The Model Card Toolkit [23] supports the creation of Model Cards and can also use metadata from ML Metadata to prepopulate information such as class distributions and performance metrics. DAG Cards [98], inspired by model cards, have also been proposed as a form of documentation, which can be automatically generated from ML pipeline code [9]. Experiment tracking tools like mlflow [115] can log metadata as a starting point for creating documentation for ML models. OpenML [108] is a popular platform for sharing datasets, ML tasks, workflows, and experimentation runs. While it supports documentation like a dataset description for dataset uploads [75], it does not enforce their quality and prioritises a frictionless user experience over documentation completeness. However, OpenML automatically analyses uploaded datasets to compute additional data quality statistics. For ML pipelines, it relies on extensions for popular libraries like scikit-learn that can automatically create a serialisable description [76]. Systems like Macaroni [55] allow querying the existing metadata in open repositories, based on a unified representation [56].

Open questions and challenges. The main drawback of model cards and datasheets is that creating and maintaining helpful documentation still mostly depends on the goodwill of the parties involved in the creation of the models and the data. Most importantly, this documentation is not machine-readable in a way that would make it easy to audit and/or verify the claims made about the provided models and data. As discussed, models are nowadays often downloaded and fine-tuned programmatically (e.g., via the popular transformers library from HuggingFace [43]). Such packages and the underlying infrastructure pose a direct opportunity to automate provenance tracking and to record the relationships between models. The semi-automated metadata collection tools can export implementation details for reproducing experiments, however, they still put the burden to extract information about the ML pipelines and models onto the users. Recently proposed approaches such as mlwhatif [29] might be a starting point to automatically extract meaningful metadata, e.g., for nutritional labels in ranking [96, 113].

Another recent trend are parameter-efficient fine-tuning methods [42, 52, 54, 59], which do not create a full model copy, but only learn a continuous prompt or an "adapter" to the model. In such cases, we would need to track provenance on the level of these prompts and adapters (which might later even be further combined [89]). A final challenge with tracking the provenance of data in generative models is that many large datasets commonly used for these models (e.g., LAION [88] or gitschemas [22]) for generative models consist of links to resources on the web, which are often crawled and filtered to build a custom dataset. This filtering process must also be taken into account for provenance.

3.3 Identifying "Contaminated" Data and Pipeline Steps Through Audits

It is still unclear how to efficiently and comprehensively audit AI/ML applications; see [13, 81] for a discussion on the current state of this endeavor. Due to our data-centric perspective, we focus on issues and directions for quantitative data audits only. As discussed in Section 2, traceback investigations in the food supply chain allow disease detectives to audit these supply chains, identify the point of contamination, and ultimately remove the source of contamination by issuing comprehensive recalls for all affected end products. How can we audit AI/ML applications in a similar manner, based on the provenance information from Section 3.2? Ideally, we would like to be able to quickly identify "contaminated" data and intermediate outputs, which, for example, contain unwanted stereotypes or has been rendered unrepresentative due to biased filtering operations. Once such contaminated data is identified, an audit would furthermore need to determine which models and predictions were affected and need to be retracted and/or recomputed. Furthermore, such data-centric audits should be able to answer a larger set of related questions about the robustness and regulatory compliance of an AI/ML application. Examples of such questions are what data and features were used by the application and whether this usage was in line with legal requirements (e.g., from the EU AI Act [20]), or whether the application follows the timely data deletion requirements imposed by the right to be forgotten from GDPR [25]. Furthermore, audits should be able to assess whether an application is robust enough against potential errors and changes in the data, and whether appropriate measures have been taken to quantify and control the fairness of its predictions.

State-of-the-art. The validation of ML data in popular ML platforms such as Google TFX [7] or Amazon SageMaker [71] relies on libraries such as Tensorflow Data Validation (TFDV) [15] and Deequ [85, 86], which generate validation rules based on heuristics and data profiling. Related approaches are to "lint" ML data based on well-known practical issues [45]. Follow-up work to these approaches [78, 91] applies a technique called "partition summarisation" to learn to spot data with potential quality issues by applying anomaly detection based on the statistics of previously observed data partitions.

There has been extensive research on cleaning datasets, e.g., [2, 46, 61, 68]. Furthermore, the data-centric AI community started developing related techniques that jointly consider the ML model and data to address inaccuracy, bias, and fragility in real-world ML applications and are tackling tasks such as training set selection and data acquisition [64]. Many of the techniques in this space rely on data influence estimation techniques [40], in particular on (an estimate of) the leave-one-out error or data Shapley value [28], which is either computed via extensive retraining or influence functions [51]. Such techniques are the basis of several recently proposed data debugging methods like Rain [111], Gopher [77] or DataScope [49]. A related line of work tackles ML pipelines and employs light-weight provenance tracking and automatic instrumentation of Python code to assess technical bias introduced by sudden distribution shifts [30, 32], data leakage and fairness issues [83, 84], as well as robustness to erroneous input data [29, 31].

Open questions and challenges. Unfortunately, neither TFDV nor Deequ have a particular focus on identifying fairness and bias issues in the data, and require a relatively high user expertise and knowledge of the underlying domain to adjust and filter the suggested validation rules. It would be crucial to find ways to guide users in designing compliance- and fairness-related data unit tests with these libraries.

Furthermore, the existing methods for estimating the influence of training samples are extremely restricted in terms of efficiency, scalability or applicability. In general, there exist two families of methods: Retraining-based methods are applicable to any model class, but require extensive retraining of the ML model on a large number of subsets of the data. Even retraining a model a few hundred times for a large dataset is infeasible in practice. The second family are gradient-based methods, which require no retraining but are only applicable to certain model classes due to assumptions of convexity [51] or linearity [114], and are still rather compute-heavy, as they often require to compute a "Hessian vector product" for each combination of a training and validation sample [40]. Some exciting progress has been made in terms of scalability, e.g., on efficiently computing the Data Shapley value [28] for kNN proxy models [47]. However, these techniques are only applicable to certain utility functions but, for example, not to common ranking-based metrics in information retrieval.

The work from the data-centric AI community is promising. However, challenges such as ML pipelines with complex data preprocessing operations are often overlooked, and automatically applying these techniques to ML pipelines is still an open challenge [33]. The approaches for the holistic screening of ML training pipelines rely on well-written code, which is often an unrealistic assumption in practice.

On the engineering side, we should strive to design a standardised API for provenance-based data auditing and incident investigation, which could be integrated into popular projects such as Google TFX, mlflow recipes, or SageMaker. Based on such an API, the academic and open source community could develop general auditing software to greatly reduce the costs of such audits.

4 Conclusion

We took a detailed look at how the FDA detects outbreaks of foodborne illnesses via their PulseNet database, discovers the contaminated food with disease detectives, and conducts traceback investigations through the food supply chain to determine the root cause of the contamination and issue a comprehensive product recall (Section 2). Inspired by the FDA's processes, we developed a technical data-centric vision for responsible AI, which centers around prediction monitoring, data tracing through end-to-end AI/ML applications, and identifying contaminated data and pipeline steps through audits. For each of these aspects, we outlined technical research ideas, reviewed related work, and discussed challenges and open questions.

We hope that our ideas can positively influence the development of safer AI/ML applications, especially in the high-risk areas outlined by recent regulation such as the upcoming EU AI act.

References

[1] Apache airflow. `https://airflow.apache.org/`.

[2] Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. Detecting data errors: where are we and what needs to be done? Proc. VLDB Endow., 9(12):993–1004, aug 2016.

[3] Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pages 298–306, 2021.

[4] California Privacy Protection Agency. California consumer privacy act - frequently asked questions.

[5] Yael Amsterdamer, Daniel Deutch, and Val Tannen. Provenance for aggregate queries. In Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 153–164, 2011.

[6] Abolfazl Asudeh, Zhongjun Jin, and HV Jagadish. Assessing and remedying coverage for a given dataset. In 2019 IEEE 35th International Conference on Data Engineering (ICDE), pages 554–565. IEEE, 2019.

[7] Denis Baylor, Eric Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, et al. Tfx: A tensorflow-based production-scale machine learning platform. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1387–1395, 2017.

[8] Denis Baylor, Kevin Haas, Konstantinos Katsiapis, Sammy Leong, Rose Liu, Clemens Menwald, Hui Miao, Neoklis Polyzotis, Mitchell Trott, and Martin Zinkevich. Continuous training for production ML in the TensorFlow extended (TFX) platform. In 2019 USENIX Conference on Operational Machine Learning (OpML 19), pages 51–53, Santa Clara, CA, May 2019. USENIX Association.

[9] David Berg, Ravi Kiran Chirravuri, Romain Cledat, Savin Goyal, Ferras Hamad, and Ville Tuulos. Open-sourcing metaflow, a human-centric framework for data science. Netflix Tech Blog, 201, 2019.

[10] Abeba Birhane, Vinay Prabhu, Sang Han, and Vishnu Naresh Boddeti. On hate scaling laws for data-swamps. arXiv preprint arXiv:2306.13141, 2023.

[11] Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1536–1546. IEEE, 2021.

[12] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. arXiv preprint arXiv:2110.01963, 2021.

[13] Abeba Birhane, Ryan Steed, Victor Ojewale, Briana Vecchione, and Inioluwa Deborah Raji. Ai auditing: The broken bus on the road to ai accountability. arXiv preprint arXiv:2401.14462, 2024.

[14] Tobias Bleifuß, Leon Bornemann, Theodore Johnson, Dmitri V. Kalashnikov, Felix Naumann, and Divesh Srivastava. Exploring change: a new dimension of data analytics. Proc. VLDB Endow., 12(2):85–98, oct 2018.

[15] Eric Breck, Neoklis Polyzotis, Sudip Roy, Steven Whang, and Martin Zinkevich. Data validation for machine learning. In MLSys, 2019.

[16] Centers for Disease Control & Prevention. PulseNet, 2024.

[17] Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? Advances in neural information processing systems, 31, 2018.

[18] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. Continual learning for generative retrieval over dynamic corpora. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, page 306–315, New York, NY, USA, 2023. Association for Computing Machinery.

[19] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Slice finder: Automated data slicing for model validation. In 2019 IEEE 35th International Conference on Data Engineering (ICDE), pages 1550–1553. IEEE, 2019.

[20] European Commission. Ai act.

[21] Emily Dinan, Gavin Abercrombie, Stevie A Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, Verena Rieser, et al. Safetykit: First aid for measuring safety in open-domain conversational systems. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2022.

[22] Till Döhmen, Madelon Hulsebos, Christian Beecks, and Sebastian Schelter. Gitschemas: A dataset for automating relational data preparation tasks. In 2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW), pages 74–78. IEEE, 2022.

[23] Huanming Fang, Hui Miao, Karan Shukla, Dan Nanas, Catherina Xu, Christina Greer, Neoklis Polyzotis, Tulsee Doshi, Tiffany Deng, Margaret Mitchell, et al. Introducing the model card toolkit for easier model transparency reporting. Google AI Blog, 2020.

[24] Batya Friedman and Helen Nissenbaum. Bias in computer systems. ACM Transactions on information systems (TOIS), 14(3):330–347, 1996.

[25] GDPR.eu. Article 17: Right to be forgotten. https://gdpr.eu/article-17-right-to-be-forgotten.

[26] GDPR.eu. Recital 74: Responsibility and liability of the controller. https://gdpr.eu/recital-74-responsibility-and-liability-of-the-controller/.

[27] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. Communications of the ACM, 64(12):86–92, 2021.

[28] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In International conference on machine learning, pages 2242–2251. PMLR, 2019.

[29] Stefan Grafberger, Paul Groth, and Sebastian Schelter. Automating and optimizing data-centric what-if analyses on native machine learning pipelines. SIGMOD, 2023.

[30] Stefan Grafberger, Paul Groth, Julia Stoyanovich, and Sebastian Schelter. Data distribution debugging in machine learning pipelines. The VLDB Journal, 31(5):1103–1126, 2022.

[31] Stefan Grafberger, Shubha Guha, Paul Groth, and Sebastian Schelter. mlwhatif: What if you could stop re-implementing your machine learning pipeline analyses over and over? Proc. VLDB Endow., 16(12):4002–4005, aug 2023.

[32] Stefan Grafberger, Shubha Guha, Julia Stoyanovich, and Sebastian Schelter. Mlinspect: A data distribution debugger for machine learning pipelines. SIGMOD, 2021.

[33] Stefan Grafberger, Bojan Karlaš, Paul Groth, and Sebastian Schelter. Towards declarative systems for data-centric machine learning. DMLR workshop @ ICML, 2023.

[34] Todd J Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 31–40, 2007.

[35] Paul Groth. Transparency and reliability in the data supply chain. IEEE Internet Computing, 17(2):69–71, 2013.

[36] The Guardian. This article is more than 4 months old Supermarket AI meal planner app suggests recipe that would create chlorine gas. `https://www.theguardian.com/world/2023/aug/10/pak-n-save-savey-meal-bot-ai-app-malfunction-recipes`, 2023.

[37] Shubha Guha, Falaah Arif Khan, Julia Stoyanovich, and Sebastian Schelter. Automated data cleaning can hurt fairness in machine learning-based decision making. In 2023 IEEE 39th International Conference on Data Engineering (ICDE), pages 3747–3754. IEEE, 2023.

[38] Shubha Guha, Falaah Arif Khan, Julia Stoyanovich, and Sebastian Schelter. Automated data cleaning can hurt fairness in machine learning-based decision making. In Transactions on Knowledge and Data Engineering (TKDE). IEEE, 2024.

[39] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, page 55–64, New York, NY, USA, 2016. Association for Computing Machinery.

[40] Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. arXiv preprint arXiv:2212.04612, 2022.

[41] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In Proceedings of the 2019 CHI conference on human factors in computing systems, pages 1–16, 2019.

[42] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. ICLR, 2022.

[43] HuggingFace. transformers package, 2024.

[44] Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. Social biases in nlp models as barriers for persons with disabilities. arXiv preprint arXiv:2005.00813, 2020.

[45] Nick Hynes, D Sculley, and Michael Terry. The data linter: Lightweight, automated sanity checking for ml data sets. In NIPS MLSys Workshop, volume 1, 2017.

[46] Sebastian Jäger, Arndt Allhorn, and Felix Bießmann. A benchmark for data imputation methods. Frontiers in Big Data, page 48, 2021.

[47] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1167–1176. PMLR, 2019.

[48] Krishna Kamath, Aneesh Sharma, Dong Wang, and Zhijun Yin. Realgraph: User interaction prediction at twitter. In user engagement optimization workshop@ KDD, number ii, 2014.

[49] Bojan Karlaš, David Dao, Matteo Interlandi, Bo Li, Sebastian Schelter, Wentao Wu, and Ce Zhang. Data debugging with shapley importance over end-to-end machine learning pipelines. arXiv preprint arXiv:2204.11131, 2022.

[50] Konstantinos Katsiapis, Abhijit Karmarkar, Ahmet Altay, Aleksandr Zaks, Neoklis Polyzotis, Anusha Ramesh, Ben Mathes, Gautam Vasudevan, Irene Giannoumis, Jarek Wilkiewicz, Jiri Simsa, Justin Hong, Mitchell Trott, Noé Lutz, Pavel A. Dournov, Robert Crowe, Sarah Sirajuddin, Tris Brian Warkentin, and Zhitao Li. Towards ML engineering: A brief history of tensorflow extended (TFX). CoRR, abs/2010.02013, 2020.

[51] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1885–1894. PMLR, 06–11 Aug 2017.

[52] Brian Lester et al. The power of scale for parameter-efficient prompt tuning. EMNLP, 2021.

[53] Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. Cleanml: A study for evaluating the impact of data cleaning on ml classification tasks. In 2021 IEEE 37th International Conference on Data Engineering (ICDE), pages 13–24. IEEE, 2021.

[54] Xiang Lisa Li et al. Prefix-tuning: Optimizing continuous prompts for generation. ACL, 2021.

[55] Ziyu Li, Henk Kant, Rihan Hai, Asterios Katsifodimos, and Alessandro Bozzon. Macaroni: Crawling and enriching metadata from public model zoos. In International Conference on Web Engineering, pages 376–380. Springer, 2023.

[56] Ziyu Li, Henk Kant, Rihan Hai, Asterios Katsifodimos, Marco Brambilla, and Alessandro Bozzon. Metadata representations for queryable repositories of machine learning models. IEEE Access, 2023.

[57] Yin Lin, Yifan Guan, Abolfazl Asudeh, and HV Jagadish. Identifying insufficient data coverage in databases with multiple relations. Proceedings of the VLDB Endowment, 13(11), 2020.

[58] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In International conference on machine learning, pages 3122–3130. PMLR, 2018.

[59] Xiao Liu et al. GPT understands, too. AI Open, 2023.

[60] Li Lucy and David Bamman. Gender and representation bias in gpt-3 generated stories. In Proceedings of the Third Workshop on Narrative Understanding, pages 48–55, 2021.

[61] Mohammad Mahdavi, Ziawasch Abedjan, Raul Castro Fernandez, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. Raha: A configuration-free error detection system. In Proceedings of the 2019 International Conference on Management of Data, SIGMOD '19, page 865–882, New York, NY, USA, 2019. Association for Computing Machinery.

[62] Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 15009–15018, 2023.

[63] The Markup. AI Detection Tools Falsely Accuse International Students of Cheating. https://themarkup.org/machine-learning/2023/08/14/ai-detection-tools-falsely-accuse-international-students-of-cheating, 2023.

[64] Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, Jessica Quaye, Charvi Rastogi, Douwe Kiela, David Jurado, David Kanter, Rafael Mosquera, Juan Ciro, Lora Aroyo, Bilge Acun, Lingjiao Chen, Mehul Smriti Raje, Max Bartolo, Sabri Eyuboglu, Amirata Ghorbani, Emmett Goodman, Oana Inel, Tariq Kane, Christine R. Kirkpatrick, Tzu-Sheng Kuo, Jonas Mueller, Tristan Thrush, Joaquin Vanschoren, Margaret Warren, Adina Williams, Serena Yeung, Newsha Ardalani, Praveen Paritosh, Lilith Bat-Leah, Ce Zhang, James Zou, Carole-Jean Wu, Cody Coleman, Andrew Ng, Peter Mattson, and Vijay Janapa Reddi. Dataperf: Benchmarks for data-centric ai development, 2023.

[65] Sean McGregor. Preventing repeated real world ai failures by cataloging incidents: The ai incident database. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 15458–15463, 2021.

[66] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In Proceedings of the conference on fairness, accountability, and transparency, pages 220–229, 2019.

[67] Mohammad Hossein Namaki, Avrilia Floratou, Fotis Psallidas, Subru Krishnan, Ashvin Agrawal, Yinghui Wu, Yiwen Zhu, and Markus Weimer. Vamsa: Automated provenance tracking in data science scripts. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pages 1542–1551, 2020.

[68] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. Can foundation models wrangle your data? Proceedings of the VLDB Endowment, 16(4):738–746, 2022.

[69] Arvind Narayanan. Fairness definitions and their politics. ACM FaccT, 2018.

[70] National Archives. Code of Federal Regulations - Traceability Plan, 2024.

[71] David Nigenda, Zohar Karnin, Muhammad Bilal Zafar, Raghu Ramesha, Alan Tan, Michele Donini, and Krishnaram Kenthapadi. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 3671–3681, 2022.

[72] Democracy Now. Meet Porcha Woodruff, Detroit Woman Jailed While 8 Months Pregnant After False AI Facial Recognition. https://www.democracynow.org/2023/8/9/porcha_woodruff_false_facial_recognition_arrest, 2023.

[73] Stephen Oladele. A comprehensive guide on how to monitor your models in production. https://neptune.ai/blog/how-to-monitor-your-models-in-production-guide, 2023.

[74] Christopher Olston, Noah Fiedel, Kiril Gorovoy, Jeremiah Harmsen, Li Lao, Fangwei Li, Vinu Rajashekhar, Sukriti Ramesh, and Jordan Soyke. Tensorflow-serving: Flexible, high-performance ml serving. arXiv preprint arXiv:1712.06139, 2017.

[75] OpenML. Dataset upload tutorial. https://openml.github.io/openml-python/develop/examples/30_extended/create_upload_tutorial.html.

[76] OpenML. Documentation. https://docs.openml.org/.

[77] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. Interpretable data-based explanations for fairness debugging. In Proceedings of the 2022 International Conference on Management of Data, pages 247–261, 2022.

[78] Sergey Redyuk, Zoi Kaoudi, Volker Markl, and Sebastian Schelter. Automating data quality validation for dynamic data ingestion. In EDBT, pages 61–72, 2021.

[79] Reza Rokni. Using tfx inference with dataflow for large scale ml inference patterns. https://blog.tensorflow.org/2021/05/using-tfx-inference-with-dataflow-for-large-scale-ml-inference-patterns.html, 2021.

[80] Svetlana Sagadeeva and Matthias Boehm. Sliceline: Fast, linear-algebra-based slice finding for ml model debugging. In Proceedings of the 2021 International Conference on Management of Data, pages 2290–2299, 2021.

[81] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. Auditing algorithms: Research methods for detecting discrimination on internet platforms. Data and discrimination: converting critical concerns into productive inquiry, 22(2014):4349–4357, 2014.

[82] Sebastian Schelter, Joos-Hendrik Böse, Johannes Kirschnick, Thoralf Klein, and Stephan Seufert. Automatically tracking metadata and provenance of machine learning experiments. In NeurIPS 2017, 2017.

[83] Sebastian Schelter, Stefan Grafberger, Shubha Guha, Bojan Karlas, and Ce Zhang. Proactively screening machine learning pipelines with arguseyes. In Companion of the 2023 International Conference on Management of Data, pages 91–94, 2023.

[84] Sebastian Schelter, Stefan Grafberger, Shubha Guha, Olivier Sprangers, Bojan Karlaš, and Ce Zhang. Screening native ml pipelines with "arguseyes". CIDR, 2022.

[85] Sebastian Schelter, Stefan Grafberger, Philipp Schmidt, Tammo Rukat, Mario Kiessling, Andrey Taptunov, Felix Biessmann, and Dustin Lange. Differential data quality verification on partitioned data. In 2019 IEEE 35th International Conference on Data Engineering (ICDE), pages 1940–1945. IEEE, 2019.

[86] Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger. Automating large-scale data quality verification. Proceedings of the VLDB Endowment, 11(12):1781–1794, 2018.

[87] Sebastian Schelter, Tammo Rukat, and Felix Bießmann. Learning to validate the predictions of black box classifiers on unseen data. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pages 1289–1299, 2020.

[88] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems, 35:25278–25294, 2022.

[89] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras, 2023.

[90] Nima Shahbazi, Nikola Danevski, Fatemeh Nargesian, Abolfazl Asudeh, and Divesh Srivastava. Through the fairness lens: Experimental analysis and evaluation of entity matching. VLDB, 2023.

[91] Shreya Shankar, Labib Fawaz, Karl Gyllstrom, and Aditya Parameswaran. Automatic and precise data validation for machine learning. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pages 2198–2207, 2023.

[92] Shreya Shankar, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, JD Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G Parameswaran, and Eugene Wu. Spade: Synthesizing assertions for large language model pipelines. arXiv preprint arXiv:2401.03038, 2024.

[93] Shreya Shankar and Aditya Parameswaran. Towards observability for production machine learning pipelines, 2022.

[94] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. arXiv preprint arXiv:1909.01326, 2019.

[95] Julia Stoyanovich, Serge Abiteboul, Bill Howe, HV Jagadish, and Sebastian Schelter. Responsible data management. Communications of the ACM, 65(6):64–74, 2022.

[96] Julia Stoyanovich and Bill Howe. Nutritional labels for data and models. A Quarterly bulletin of the Computer Society of the IEEE Technical Committee on Data Engineering, 42(3), 2019.

[97] Ki Hyun Tae, Yuji Roh, Young Hun Oh, Hyunsu Kim, and Steven Euijong Whang. Data cleaning for accurate, fair, and robust models: A big data-ai integration approach. In Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning, pages 1–4, 2019.

[98] Jacopo Tagliabue, Ville Tuulos, Ciro Greco, and Valay Dave. Dag card is the new model card. DCAI workshop @ NeurIPS, 2021.

[99] Wang Chiew Tan et al. Provenance in databases: Past, current, and future. IEEE Data Eng. Bull., 30(4):3–12, 2007.

[100] Ars Technica. UnitedHealth uses AI model with 90% error rate to deny care, lawsuit alleges. `https://arstechnica.com/health/2023/11/ai-with-90-error-rate-forces-elderly-out-of-rehab-nursing-homes-suit-claims/`, 2023.

[101] The Intercept. Facebook Engineers: We Have No Idea Where We Keep All Your Personal Data, 2022.

[102] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE transactions on pattern analysis and machine intelligence, 30(11):1958–1970, 2008.

[103] DigiChina Stanford University. Internet information service algorithmic recommendation management provisions.

[104] U.S. Food & Drug Administration. Food Traceability List, 2024.

[105] U.S. Food & Drug Administration. Frequently Asked Questions: FSMA Food Traceability Rule, 2024.

[106] U.S. Food & Drug Administration. Outbreak of Salmonella Newport Infections Linked to Onions, 2024.

[107] U.S. Food & Drug Administration. Outbreaks of Foodborne Illness, 2024.

[108] Jan N. van Rijn, Bernd Bischl, Luis Torgo, Bo Gao, Venkatesh Umaashankar, Simon Fischer, Patrick Winter, Bernd Wiswedel, Michael R. Berthold, and Joaquin Vanschoren. Openml: A collaborative science platform. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, Machine Learning and Knowledge Discovery in Databases, pages 645–649, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[109] Maria Vechtomova. Traceability & reproducibility. `https://marvelousmlops.substack.com/p/traceability-and-reproducibility`, 2023.

[110] Steven Euijong Whang, Ki Hyun Tae, Yuji Roh, and Geon Heo. Responsible ai challenges in end-to-end machine learning. arXiv preprint arXiv:2101.05967, 2021.

[111] Weiyuan Wu, Lampros Flokas, Eugene Wu, and Jiannan Wang. Complaint-driven training data debugging for query 2.0. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pages 1317–1334, 2020.

[112] Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. Bot-adversarial dialogue for safe conversational agents. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2950–2968, 2021.

[113] Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, HV Jagadish, and Gerome Miklau. A nutritional label for rankings. In Proceedings of the 2018 International Conference on Management of Data, SIGMOD '18, page 1773–1776, New York, NY, USA, 2018. Association for Computing Machinery.

[114] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. Advances in neural information processing systems, 31, 2018.

[115] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. Accelerating the machine learning lifecycle with mlflow. IEEE Data Eng. Bull., 41(4):39–45, 2018.