Navigating Data-Centric Artificial Intelligence
With DC-Check: Advances, Challenges, and Opportunities

Nabeel Seedat ©, Fergus Imrie ©, and Mihaela van der Schaar ©, Fellow, IEEE

Abstract—Data-centric artificial intelligence (AI) is an emerging paradigm that emphasizes the critical role of data in real-world machine learning (ML) systems—as a complement to model development. However, data-centric AI is still in its infancy, lacking a standardized framework that outlines necessary data-centric considerations at various stages of the ML pipeline: Data, Training, Testing, and Deployment. This lack of guidance hampers effective communication and design of data-centric driven ML systems. To address this critical gap, we introduce the Data-Centric Checklist (DC-Check), an actionable checklist-style framework that encapsulates data-centric considerations for ML systems. DC-Check is aimed at both practitioners and researchers to serve as a reference guide to data-centric AI development. Around each question in DC-Check, we discuss the applicability of different approaches, survey the state of the art, and highlight specific data-centric AI challenges and research opportunities. While developing DC-Check, we also undertook an analysis of the current data-centric AI landscape. The insights obtained from this exploration support the DC-Check framework, reinforcing its utility and relevance in the rapidly evolving field. To make DC-Check and related resources easily accessible, we provide a DC-Check companion website (https://www.vanderschaar-lab.com/dc-check/), which will serve as a living resource, updated as methods and tools evolve.

Impact Statement—Data-centric AI has been raised as an important paradigm shift to improve the reliability AI systems—giving data a central stage compared to model-centric approaches. However, there is currently no standardized process or guide to the necessary considerations for data-centric AI systems, making the agenda hard to engage with practically. This article aims to solve this and facilitate ease of engagement by framing the article within a checklist-style framework called DC-Check, providing actionable data-centric questions and considerations for researchers and practitioners. For each question, DC-Check provides an overview of current data-centric tools and modeling approaches, while also including research opportunities. Finally, while our focus is the current state of data-centric AI, DC-Check was designed with longevity in mind. Hence, while data-centric tools will certainly evolve, the DC-Check checklist represents the fundamental questions and considerations of data-centric AI more generally. Since these questions represent pertinent data-centric considerations, irrespective of the state of tooling, we believe this makes it more future-proof to stand the test of time.

Index Terms—Data-centric artificial intelligence (AI), machine learning (ML) pipelines, reliable-ML.

I. INTRODUCTION

REAL-WORLD machine learning: Machine learning (ML) has seen numerous algorithmic breakthroughs in recent years. This has increased the appetite for adoption across a variety of industries: from recommendation systems in e-Commerce [1], speech recognition for virtual assistants [2], natural language processing for machine translation [3], computer vision for medical image analysis [4], and a deluge of tabular data applications across healthcare [5], finance [6], and manufacturing [7]. In realizing these leaps, the focus in the ML community has often been to optimize algorithms for state-of-the-art (SOTA) performance on benchmark datasets, whilst neglecting a swathe of considerations necessary in real-world settings including, but not limited to: data curation, data quality assessment and cleaning, characterizing subgroups in data, model robustness, monitoring and accounting for data drifts, and much more. This gap to reality has meant that real-world systems, with notable exceptions, have been unable to unequivocally and consistently perform as expected.

Failure of ML: In fact, there have been numerous high-profile ML failures such as gender and racial biases in Twitter’s image cropping algorithm [8], football object tracking algorithms latching onto spurious correlations and mistaking the linesman’s head for the ball [9], or Google Health’s diabetic retinopathy system, which failed on images with lower quality than development [10]. Furthermore, approximately 85% of industrial artificial intelligence (AI)/ML systems are projected to “deliver erroneous outcomes due to bias in data or algorithms,” while only 53% bridge the gap between prototypes and production [11]. This highlights that simply deploying a seemingly highly predictive model is but one piece of the puzzle for a reliable ML system in practice. In fact, a recurring theme across anecdotes is the data being a driver of these failures.
In this work, we outline why the data work, often undervalued by the ML community is often a source of real-world brittleness. Conversely, in model-centric AI, the data are considered merely operational, is key to unlocking reliable ML systems. Consequently, our vision is a systematic and principled framework that can be adopted in diverse, high-stakes, and complex domains, such as healthcare, finance, and law enforcement. Consequently, we believe that a systematic and principled framework that can be used to guide the development of end-to-end ML pipelines is sorely needed.

Rethinking the focus of ML: Specifically, we propose that the nascent area of “data-centric AI” should serve as a harmonizing umbrella. Data play a critical role in ML, with its characteristics and quality influencing each stage of the pipeline. Consequently, our vision is a systematic and principled framework that places “data at the center.”

Contrasting paradigms: At this point, it is pertinent to define and contrast current notions of data-centric AI vs model-centric AI (the current de facto standard) [28], [29], [30], [31]. Data-centric AI views model or algorithmic refinement as less important (and in certain settings algorithmic development is even considered as a solved problem), and instead seeks to systematically improve the data used by ML systems. Conversely, in model-centric AI, the data are considered an asset adjacent to the model and are often fixed or static (e.g., benchmarks), whilst improvements are sought specifically to the model or algorithm itself. We believe that the current focus on models and architectures as a panacea in the ML community is often a source of real-world brittleness. In this work, we outline why the data work, often undervalued as merely operational, is key to unlocking reliable ML in the wild.

**Related efforts:** The ML community has attempted to improve the usage of ML systems with machine learning operations (MLOps) [12], [13], [14], robust machine learning [15], [16], [17], [18], and trustworthy ML [19], [20], [21]. While individually each addresses important yet different challenges, in isolation, none are sufficient as a “silver bullet” for reliable ML at the end-to-end ML pipeline level, i.e., covering the ML lifecycle from data all the way to deployment. Such considerations are especially relevant as ML systems are increasingly adopted in diverse, high-stakes, and complex domains, such as healthcare, finance, and law enforcement. Consequently, we believe that a systematic and principled framework that can be used to guide the development of end-to-end ML pipelines is sorely needed.

**Extending DCAI definitions:** That said, in the current form, definitions of data-centric vs model-centric are insufficient when considering end-to-end pipelines. A limitation is the binary delineation of working only on the data or model. In this work, we go further and call for an expanded definition of data-centric AI such that a data-centric lens is applicable for end-to-end pipelines.

**Need for standardization:** While data-centric AI is still in its infancy, it has been raised as an important concept and paradigm shift (from model-centric AI) to improve the reliability of ML systems [28], [29], [30], [31]. However, currently there is no standardized process in which to communicate the design of data-centric ML pipelines. More specifically, there is no guide to the necessary considerations for data-centric AI systems, making the agenda hard to engage with practically.

**DEFINITION:** DATA-CENTRIC AI

Data-centric AI encompasses methods and tools to systematically characterize, audit, generate, and monitor the underlying data used to train and evaluate ML models.

At the ML pipeline level, this means that the considerations at each stage should be informed in a data-driven manner.

We term this a data-centric lens. Since data are the fuel for any ML system, we should keep a sharp focus on the data, yet rather than ignoring the model, we should leverage the data-driven insights as feedback to systematically improve the model.

To address this gap and facilitate ease of engagement, we have structured our outline of data-centric AI through an actionable checklist style framework, which we refer to as DC-Check. The framework advocates for a data-centric lens encompassing the following four stages of the ML pipeline (see Fig. 2).

The checklist consists of a series of important questions and considerations necessary for data-centric AI, and reliable ML systems more generally— which are summarized in Fig. 2.
Hence, while methods and tools will naturally evolve, we believe that the checklist’s questions will remain a pertinent reference when navigating data-centric AI.

1) Data: Considerations to improve the quality of data used for model training, such as proactive data selection, data curation, and data cleaning.

2) Training: Considerations based on understanding the data to improve model training, such as data-informed model design, domain adaptation, and robust training.

3) Testing: Considerations around novel data-centric methods to test ML models, such as informed data splits, targeted metrics and stress tests, and evaluation on subgroups.

4) Deployment: Considerations based on data postdeployment, such as data and model monitoring, model adaptation and retraining, and uncertainty quantification.

The role of ML process documentation: The ML community has begun to engage with similar concepts of documentation for ML processes. In particular, Datasheets for Datasets [32] propose documentation guidelines for new datasets, while Model Cards for Model Reporting [33] advocates for standardized reporting of models, including training data, performance measures, and limitations. We consider these documentation paradigms as complementary to DC-Check, as first, they both only consider one specific part of the pipeline (data and model, respectively), and second, their focus is on improving documentation after the fact. In contrast, DC-Check covers the entire pipeline and advocates for engaging with these considerations before building ML systems. We hope that DC-Check aimed at both practitioners (ML/software engineers and data scientists) and researchers will encourage thoughtful data-centric decision-making from the offset regarding the impact of data and its influence on the design of ML systems at each stage of the ML pipeline.

The role of ML process documentation: The article is structured as follows. Each of the four components (Data, Training, Testing, and Deployment) is divided into a set of data-centric questions, to guide users to understand potential challenges. From the practical overview perspective, for each question, we survey the current status quo of related approaches, as well as, specific SOTA tools that could be used to address each question. In addition to the checklist that guides projects, each question includes research opportunities to advance the nascent research area of data-centric AI.

Going beyond a documentation tool, DC-Check supports practitioners and researchers to achieve greater transparency and accountability with regard to data-centric considerations for ML pipelines. We believe that this type of transparency and accountability provided by DC-Check can be useful to policymakers, regulators, and decision makers to understand the design considerations at each stage of the ML pipeline.

Worked examples and example tooling: To guide usage, we provide worked examples of DC-Check for two real-world projects in the Appendix, along with example tooling for each component of DC-Check at the end of the sections in the main article (Tables I–IV) Finally, before delving into DC-Check, we highlight that the relevant data-centric considerations will naturally differ based on the “type” and “level” of reliability required. In particular, different applications have different stakes and this will inform which aspects of DC-Check are
most relevant. As a consequence, the DC-Check questions are by no means prescriptive, complete, or exhaustive. We expect that certain questions will be more relevant depending on the existing workflows, context, stakeholders, or ML use cases.

II. DATA: A DATA-CENTRIC LENS INFORMING DATASET CURATION

The performance and robustness of ML systems often depend on the characteristics and quality of data [34], [35], [36]. In particular, data with imperfections or limitations can lead to suboptimal model performance. Consequently, systematic data curation is crucial to ensure ML reliability. Such imperfections could arise for reasons including, but not limited to:

1) Poor quality data: data are noisy and corrupted or labels are ambiguous or mislabeled;
2) Biased data: collection process introduces bias or specific groups have coverage imbalances;
3) Missing data: specific features are missing either at random or not at random.

Such data issues result in so-called “data cascades” [36], where problems with the data affect downstream tasks. To mitigate this, the “data work” usually begins with exploratory data analysis (EDA) [37], [38], [39], [40]. Based on the EDA, the next steps involve data cleaning and curation, which involves a large amount of human effort, often performed in an ad hoc manner. Not only is the approach prone to human variability, but also dataset curation and preparation are tedious and time consuming, typically accounting for 80% of the time in a data science project [41], [42]. To address the highlighted issues and curate high-quality datasets, the goal is to empower rather than replace data scientists/ML engineers by making processes more systematic, making processes less prone to human variability, and freeing up human experts for other tasks. The goal of DC-Check’s questions for the “Data” phase is to make the data curation processes more systematic and less prone to human variability. This is summarized in Fig. 3.

Example tools for Data questions: Table I outlines example tools that could be used to address each question under “Data.”

Q1: How did you select, collect, or curate your dataset?
ML datasets are often curated or selected in a one-off “benchmarking” style fashion. We aim to raise awareness that, without careful consideration, this approach has the following potential pitfalls: 1) fixation on performance on these datasets, which might not translate to real-world performance [62]; 2) these datasets are imperfect and have labeling errors, inconsistencies, and biases [32], [53], which can influence the learning algorithm (especially true if labels are generated via crowdsourcing); and 3) once-off collection does not represent the real world. These three issues mean that, while model performance on these datasets is laudable, it does not provide real-world guarantees of performance, even for the same task. Cases such as degradation from ImageNet V1 to V2 highlight this issue, even in controlled settings [63].

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
We do not suggest that curated datasets should be completely discarded. On the contrary, they have contributed greatly toward model advancements and are a good starting point for many applications. Instead, when constructing and curating datasets, these pitfalls should be considered and addressed.

A further consideration is that high-quality datasets in academia or industry are often collected “one-off,” requiring huge amounts of human labeling effort. However, in real-world settings, models need to be frequently updated with new data (potentially daily). Even if the problem might seem static (e.g., detecting pneumonia in an x-ray), there is still a need for data updates. This raises the challenge of how to systematically curate new datasets over time, without the need for expensive human labeling. We discuss potential data-centric opportunities later to address this.

Beyond simply curating a high-quality, consistent labeled dataset, we draw attention to the need for forensics on dataset selection and what qualities make a dataset useful for a specific task. Specifically, this concerns whether the data are indeed pertinent and sufficiently representative. In particular, we want to avoid situations similar to the early stages of the Covid-19 pandemic [64], where researchers unknowingly used a common chest scan dataset from [65] as a control group against actual Covid-19 positive scans. However, as many did not realize, the control dataset consisted of pediatric patients. Hence, while there was an illusion of reliable performance in detecting Covid-19, the reality is that the data mismatch transformed the model into an adult vs child detector [66]. While seemingly an extreme case, we believe that similar scenarios such as [64] are entirely avoidable with a data-centric lens from the offset, wherein datasets are audition for such subgroup dependencies and interactions. Although data assessment to prevent this would be largely manual right now, we present potential opportunities to improve this process.

Related to forensics are considerations of bias introduced by humans through the data collection process, but also how systems themselves can bias data through degenerate feedback loops [67]. For example, a system’s output, such as a recommendation system, can influence the user’s behavior. The new behavior can then contribute to “degeneracy” in the new training dataset. Not accounting for this subtlety in how biases can be injected and insufficiently auditing them from the offset can lead to unintended behaviors. We believe that concepts of data provenance could play a role in this regard. Next, we highlight opportunities to address these challenges.

Opportunities: First, continuous dataset curation without large amounts of human labeling. Advances in weak supervision [68] or even semisupervised labeling functions based on automatic labeling, coupled with rule-based systems as has been used in NLP could allow this realization [69]. Second, merging data sources, from multiple different sources, e.g., multimodal ML (e.g., fusing images with text) or fusing disparate datasets with similar signal, e.g., datasets from different countries or datasets with different features for the same modality. Third, automated data forensics, to prevent unintended consequences for ML models. This could involve identifying inappropriate data subsets or degenerate data. While this is a currently manual process, there exist opportunities if not to automate this, but to develop human-in-the-loop methods to identify candidate problematic data.

Q2: What data cleaning and/or preprocessing, if any, has been performed? Typical steps of the colloquial term “data wrangling” and cleaning [70] could involve one or more of the following (for brevity we assume the data engineering pipeline and quality control has been accounted for): 1) data cleaning and preprocessing: obvious errors such as outliers, mislabeled or implausible data due to system level errors [71], [72] (i.e., data engineering); 2) feature selection [73], [74] and engineering, either via intuition or domain knowledge (such as including key clinical predictors or business features); and 3) data imputation of missing data [46], [75], [76]. Failure to consider such data issues can, at best, lead to a suboptimal model and, at worst, capture the incorrect statistical relationship (e.g., [22]).

Opportunities: The aforementioned processes are largely manual and rely on human intuition. Similar to how AutoML [77], [78], [79] has automated model-centric optimization, there exists opportunities to build tools to make the data-centric process more autonomous and systematic. For instance, automated data-centric (AutoDC) tools or reinforcement learning agents which select the components of such processing pipelines are potential avenues ripe for exploration.

Q3: Has data quality been assessed? The quality of the dataset affects our ability to learn. How can we quantify signal at a dataset level, especially in determining upper bounds on expected performance? Specifically, how to identify high-quality samples that are “easy” to learn from and differentiate from more challenging or lower-quality samples where additional features might be required, the statistical relationships may not hold, or the sample might be mislabeled [48]. Applicable methods include those that assess intrinsic instance hardness [50], analyzing how individual examples evolve differently based on model training dynamics [47], [48], [49] or explicitly modeling the value of examples [51].

Opportunities: Current approaches to assessing data quality can characterize samples in a dataset as easy, ambiguous, or hard. However, most methods are model-dependent (even for models that perform similarly). Consequently, the model itself might influence the characterization of individual data samples. Ideally, we desire a data-centric approach where the findings are inherent to the difficulty of the data itself, rather than the model used to assess the data (of course, assuming the models are similarly performing). The importance is to ensure that the instances identified represent the inherent difficulty with the data (aleatoric) rather than the model’s challenge with the data (epistemic) [18], [80]. The utility of this task, beyond assessing the quality of data samples, is to either sculpt the dataset by filtering the challenging data points or rigorously test performance on these challenging data points/subgroups. Moreover, beyond the identification and filtering of samples, there exists opportunities to utilize ideas from explainable AI to better understand the types of samples identified within these subsets. Current approaches provide explanations via prototypes of the identified groups [30] or via natural language [81] in the case where
cross-modality embeddings are available. There exist opportunities to improve the explanation of data quality from an algorithmic as well as human-centric perspective.

**Q4: Have you considered synthetic data?** Synthetic data generation [82] typically involves generative models to synthesize data, often considered for privacy-preserving reasons or due to data access limitations [60], [83], especially in regulated settings (e.g., healthcare or finance). Thus, synthetic data might be considered if such considerations are relevant. For situations where this is not the case, we ask if synthetic data could provide utility—for this, we detail opportunities. We note the difference between synthetic data generation and data augmentation. Synthetic data refer to new data points that did not exist in the dataset, whilst data augmentation is a new version of an existing data point.

**Opportunities:** Opportunities for synthetic data lie beyond access and privacy issues, e.g., improving data quality by increasing dataset coverage. Alternatively, generating fair and unbiased datasets reflecting the desired data generating process, which might differ from reality. Finally, generating data of specific scenarios we want to stress test.

**III. TRAINING: A DATA-CENTRIC LENS INFORMING MODEL TRAINING**

How can we ensure reliable model training? While this might sound inherently model-centric as we modify the model training, we motivate the role of data in informing model training. For instance, by understanding the data, one might highlight the need for robust training procedures. In addition, specific remedies often require knowledge of the desired properties and data distributions. Clearly, this is different from conventional model-centric notions of improving the architectures, changing model layers, or tuning hyperparameters. Although, these remain important concepts. The goal of DC-Check’s questions for the “Training” phase is to link understanding of the data to guide model building and training decisions. This is summarized in Fig. 4.

**Example tools for Training questions:** Table II outlines example tools that could be used to address each question under “Training.”

**Q5: Have you conducted a model architecture and hyperparameter search?** There is “no free lunch” when it comes to selecting a ML model for a particular task. Thus, it is important to compare a variety of models, including “simple” alternatives. We could also consider how the data could inform the architecture choice. For example, attention layers work well in NLP due to the nature of text data. Going beyond model search, documenting hyperparameter search is also crucial for reliable ML pipelines. We highlight that such search could also be done using AutoML frameworks [77], [78]. Interesting avenues include the search being over different pipelines [79], not just the models hyperparameters. Additionally, there exists opportunities where understanding the data can help to refine the search to a more limited search space.

**Opportunities:** Understanding the data could better inform the model architecture itself, allowing one to incorporate inductive biases [102] using causal models, domain knowledge, and expert models. Such understanding could also guide the hyperparameter search to make it less exhaustive, e.g., hyperparameters translating across similar datasets.

**Q6: Does the training data match the anticipated use?**

Thinking critically about the possible environments in which the model will be deployed, and whether the training data are sufficient, is important. Consider the case where the domain or distribution differs from the training data. Failure to consider
such data issues can lead to trained models with poor generalization capability or providing biased estimates. To address this in cases where it is not possible to curate or select more appropriate training data, one could consider transfer learning [103], [104] and domain adaptation [92], [103], [105], where the goal is to adapt the model or representation space to match the desired use.

Opportunities: The aforementioned methods are often predicated on the assumption that data from the target domain are readily available. Hence, opportunities exist to develop methods to reduce the reliance on data from a target domain—i.e., domain adaptation or transfer learning with limited data from the target domain. This could be useful, for example, in low-resource healthcare settings. Another area to consider is transfer learning. Although these methods are often used for "unstructured" data such as images and text, there exist opportunities to formalize how we can transfer in tabular settings, in particular, in cases with feature mismatch.

Q7: Are there different data subsets or groups of interest? There are often specific groups of interest (e.g., high-risk patients). Alternatively, we might want to ensure fairness and/or robustness across different subsets of data. Such considerations are often important in high-stakes decision making settings, such as healthcare or finance. With such considerations in mind, one could consider employing training methods with fairness objectives [106], [107] or methods that directly optimize to achieve groupwise performance parity [17], [93], i.e., instead of empirical risk minimization to maximize average performance, the goal is equalizing performance across groups, or minimizing the worst group error.

Opportunities: Current methods for optimizing groupwise parity typically need group labels, which are not always available in the data. Hence, there are opportunities to develop data-centric methods to identify such regions or subsets of data when group labels are unavailable. Additionally, these methods typically come at the cost of sacrificing overall accuracy. Novel methods should be explored to mitigate the loss of overall performance. For example, as outlined by [108], addressing fairness in subsets could harm accuracy and vice versa; improving accuracy could reduce fairness. Novel methods to better balance tradeoffs are of great need.

Q8: Is the data noisy, either in features or labels? Collected data points might be noisy and/or mislabeled. This is especially true where data labels are obtained through crowd-sourcing or automated methods. Identifying the presence of such artifacts can then motivate the use of methods to learn with noisy data. These could include noise robust loss functions, including architecture layers robust to noise, metalearning, or dataset sample selection [15].

Opportunities: A myriad of methods aim to solve the noisy data learning problem (see [15]) and, of course, can be further improved. In addition, a significant opportunity lies in the data-centric task of identifying and quantifying that indeed the dataset consists of noisy or mislabeled samples, and hence the aforementioned methods are required.

IV. TESTING: A DATA-CENTRIC LENS INFORMING NEW APPROACHES TO TESTING

Before deploying trained ML models, they need to be rigorously tested. Mature industries involve standard methods to benchmark and provide characterizations of components under a variety of testing and/or operating conditions [32]. For instance, automobiles make use of wind tunnels and crash tests to assess specific components, while electronic component datasheets outline conditions where reliable operation is guaranteed. On the contrary, current approaches to characterize the performance of ML models do not have the same level of detail and rigor. The current de facto in ML is benchmarking style, where the model is evaluated on a held-out test set and assessed on average/aggregate performance. The goal of DC-Check’s questions for the "Testing" phase is to guide better and more rigorous ML testing. This is summarized in Fig. 5.

Example tools for Testing questions: Table III outlines example tools that could be used to address each question under “Testing.”

Q9: How has the dataset been split for model training and validation? Thoughtful and careful documentation of data splits for training and validation are often overlooked, e.g., random splits might not always be optimal, especially if the curated datasets might themselves not be IID. This can happen when datasets are collected over time with seasonality or evolving patterns. Blindly aggregating the data and ignoring the temporal nature when splitting the data can at best mask phenomena and at worst mean that we might capture idiosyncrasies over time. Addressing this issue is obvious when temporal data are available. However, certain datasets such as image datasets often do not contain this information, which
might mask such phenomena. Another issue surrounding data splitting is whether subgroups are actually considered. Random splits might inadvertently introduce selection biases or coverage gaps. Assessing representativeness in data selection and splitting is vital to mitigate such issues. Often careful consideration should be given to the split used depending on both the specific problem and the use-case for the model. For example, in drug discovery, scaffold splits, where structurally different molecules are kept separate might be necessary for one use-case, but less appropriate for another [126]; similarly, cross-target splitting is necessary for models intended to be deployed on novel proteins, but less important for other applications [127].

Opportunities: Automatic methods that characterize/identify and subsequently account for subgroups are an avenue for future work. Especially in terms of assessing the representativeness of data splits. Of course, methods such as cross-validation might mitigate such issues in small cases, but new methods are needed to scale to large datasets.

Q10: How has the model been evaluated (e.g., metrics & stress tests)? Related to the issue around using benchmark datasets, we stress the importance of both how and on what data the model will be evaluated. Typical approaches might use predefined train-test splits (benchmark) or randomly split the dataset. However, we advocate for additional scenario-based evaluation, i.e., test scenarios, motivated by real-world considerations that probe specific aspects of model performance. For example, face recognition could assess performance for different scenarios such as demographic and lighting conditions. Identifying model failure scenarios could guide where additional targeted data collection might help to improve the model. Additionally, ML models are typically evaluated on average performance have their place, can we better evaluate models at a granular level? Naturally, we could either partition an existing dataset that might suffer from small sample sizes or collect data across different scenarios, which is labor-intensive.

Consequently, we draw attention to the role that synthetic data could play to generate data to stress test specific scenarios, without having to collect data from said scenarios, e.g., testing the model’s robustness under covariate shift.

In terms of subgroup characterization, current approaches rely on the assessment of specific, well-known subgroups in the data. However, what if a subgroup is not well-defined or is intersectional, yet we still wish to identify the underperforming slice. There are obvious subgroups we can assess (e.g., race and sex). However, a research challenge is to identify subgroups that perform poorly and are not tied to well-known features or might have intersectional factors. Hence, opportunities exist to develop data-centric methods for robust and automatic subgroup characterization (slice-discovery) as a way to systematically assess model performance on different regions of the data.

Opportunities always exist for new metrics, since there isn’t one metric to rule them all. From a data-centric lens, this could be calibrating metrics to the difficulty of the dataset, or the subset evaluated over. Additionally, work is needed on better proxies for what humans measure (e.g., business KPI).

V. DEPLOYMENT: A DATA-CENTRIC LENS INFORMING RELIABLE AND OBSERVABLE DEPLOYMENT

We can only realize the value of predictive models once they are deployed into the wild. This then evokes additional considerations to ensure reliability. We do not consider systems and infrastructure challenges, typically covered under the umbrella of MLOps, software engineering, and DevOps. Instead, we focus on ML-specific considerations.

First, we wish that models provide reliable predictions when operated under diverse conditions. Additionally, we expect that models “know what they do not know,” especially with uncertain or out-of-distribution (OOD) data. Second, the world is not stationary, and hence models will naturally degrade over time. Thus, one might want to include monitoring and observability to data and distribution shifts to ensure system reliability.

A data-centric lens at deployment time consists of characterizing incoming data, in order to assess if the model has the desired capabilities to make reliable predictions on said data. This brings up the conundrum, where the data have changed, of both adapting the data and subsequently retraining the model. This step closes the feedback loop between Data and Training in a data-informed manner. The goal of DC-Check’s questions for
the “Deployment” phase is guide thinking around reliability once the ML system is deployed and interacts with data in the real-world. This is summarized in Fig. 6.

Example tools for Deployment questions: Table IV outlines example tools that could be used to address each question under “Deployment.”

Q11: Are you monitoring your model? While seemingly obvious, establishing a systematic process to monitor a deployed model is a crucial but often overlooked step. Monitoring can manifest in two main ways: 1) detecting divergences in the data; and 2) mechanisms to detect the failures of the model (ideally with statistical guarantees). Monitoring the deployment distribution is crucial, as training and deployment distribution mismatch is a significant cause of ML system failure. The result is models behaving unpredictably or failing silently and issuing incorrect, yet overconfident predictions, making them challenging to audit. Hence, monitoring and observability solutions could consider methods to detect dataset shift and methods to address the shift. However, before examining each, we define the three most prevalent shifts [157]. Further, while shifts might seem like steplike, instantaneous events, the rate of shift can vary in reality. This variability impacts our ability to detect it, with gradual shifts more difficult to characterize.

1) Covariate shift: \( P(X) \) changes, \( P(Y|X) \) remains the same, e.g., diabetic retinopathy screening: training data has high quality images. However, when deployed there is a covariate shift with images captured of lower quality.

2) Label shift: \( P(Y) \) changes (i.e., prior shift), \( P(Y|X) \) remains the same, e.g., we train an image classification model to distinguish cats and dogs on North American animals. When used in Africa the distribution of target animals would have shifted.

3) Concept drift: \( P(Y|X) \) changes, \( P(X) \) remains the same, e.g., predicting customer spend. Covid-19 then changed customer behavior, affecting the statistical properties of the input–output relationship (i.e., posterior shift).

The most basic monitoring procedure is to monitor the model’s performance metrics, compared to expected performance or a baseline, i.e., service level objective, such as accuracy. However, in many real-world scenarios, ground truth labels are often either unavailable or have time lags in availability, which necessitate alternative monitoring approaches.

One solution is to monitor the data pipeline by data schema validation or placing constraints on the data values [158]. Another approach is data drift detection, which monitors the input or predicted label distributions and assesses if there is a discrepancy from the training data. One could consider simple statistics approaches to more advanced distributional divergence methods.

The aforementioned approaches assume a single train and single test set comparison, i.e., this simplifying assumption distills distribution shift as a comparison of two finite sets of data. In reality, an ML model is deployed and makes predictions over time. Hence, we need to assess drift in a streaming setting with respect to a specific time horizon. Thus, selecting the time horizon width to segment and compute potential drift is nontrivial. One could consider whether the perceived drift is likely to be gradual or rapid in determining an optimal slicing interval.

Opportunities: Approximating model performance without labels (e.g., due to lag) is an impactful opportunity. If this is
not possible, we still want to detect drift. Current methods are largely suited to low-dimensional data, and hence new methods tailored to high-dimensional settings (both tabular and nontabular) are needed. Finally, principled approaches to optimal slicing intervals would improve streaming detection.

Q12: Do you have mechanisms in place to address data shifts? Once data shift has been detected, one could consider methods to address the shift, such as adapting the model. A naive approach might attempt to train the model on a sufficiently large dataset in the hope of covering the entire distribution which we might encounter in the wild. This has easy points of failure if the full distribution is not truly captured and is only applicable in limited scenarios where large datasets are readily available. The second approach previously discussed under training is learning domain-invariant representations. This approach is one example of a data-centric lens informing model-centric development. Finally, we could adapt or retrain the model using data from the new distribution. However, there are a number of factors to consider: how can we efficiently construct a new labeled dataset, when is the optimal time to retrain, how often should a model be updated (i.e., cadence), and whether to use a batch-training approach vs continual learning approach. Answering these questions efficiently is covered in the opportunities section.

Opportunities: An overlooked opportunity is how to construct new datasets for retraining. Currently, new datasets are constructed manually and are also done so without failures informing the dataset updates. We propose feedback-driven datasets, with a data-centric enabled feedback loop. By this, we mean errors informing the automatic creation, updating, or augmentation of new training sets. In fact, the type of “self-tuning” could be informed in a data-centric manner by the type of drift, e.g., concept drift requires retraining on the latest data, whilst with covariate shift, we could augment or synthetically generate certain subsets. Additionally, methods to automatically determine the optimal retraining intervals would be hugely impactful in industrial settings where models are updated on a daily/weekly cadence.

Furthermore, what happens when many features are flagged as having drifted? It is often practically unactionable in such cases. For example, if tens or hundreds of features have high KL divergence values. Hence, new methods to actionally identify the root cause of drift are opportunities for development. Furthermore, we wish that the feedback is human understandable with recourse beyond simply highlighting features that have changed.

Q13: Have you incorporated tools to engender model trust? For models to produce trustworthy predictions, we desire that models reflect their uncertainty when issuing predictions on certain data, as well as identify when they are required to predict on data outside their realm of expertise. One could consider the myriad of uncertainty quantification techniques (e.g., Bayesian neural networks [140], conformal prediction [143], [144], Gaussian processes [142], and MC dropout [139]) to ascertain a model’s predictive uncertainty. Uncertainty quantification could then be used to defer predictions if they are too uncertain—ensuring reliability by selective prediction. Uncertainty can also provide actionable feedback on whether to trust predictions. Beyond uncertainty quantification there are other relevant areas. Methods from explainability (XAI) (e.g., feature importance, example-based, and concept-based) can be used to trace model predictions back to the data, which could both aid with user understanding but potential updates to the data itself. Further, algorithmic fairness [156] (e.g., equalized odds postprocessing [159]) could arguably also fall into this category to engender trust in a deployed model. Finally, models should identify when asked to predict on data outside its training distribution. The OOD detection can play a role in this case (e.g., likelihood ratios [151], distance-based [153], and density-based [160]), However, even if a data point lies within the support of the distribution, it could lie in a region where a model might make poor predictions. Recent work such as [30] could identify such inconsistent and incongruous examples of in-distribution data.

Opportunities: There are opportunities for improved methods of uncertainty quantification, model explainability, algorithmic fairness, and OOD detection. Additionally, not all errors are created equally, with some errors more harmful than others. Ideally, we want to assess if a model should be trusted uniformly or whether specific data subgroups have greater uncertainty, have different model explanations, are unfairly prejudiced, or contain more OOD samples.

VI. GOING BEYOND SUPERVISED LEARNING WITH DC-CHECK

The DC-Check framework is intended to be usable across a variety of contexts, use-cases, and stakeholders. Thus, while DC-Check has been framed in the context of supervised learning, it is of course not the only applicable setting.

We believe that the scope of issues and opportunities are flexible enough and arise across a wide variety of ML use-cases, from unsupervised learning to areas such as causal inference. Of course, the Data area is the most natural fit when moving beyond supervised learning.

In particular, systematic methods to curate and assess datasets are applicable across all strands of ML. Training naturally is more field- or domain-specific. However, for Deployment and Testing much more work is required to develop better methods and practices to translate beyond supervised learning.

Overall, we wish to highlight that the broad data-centric considerations and opportunities posed in DC-Check are nevertheless important for both practitioners and researchers alike in developing reliable end-to-end ML pipelines, no matter the ML paradigm.

VII. BRIDGING THE RELIABILITY GAP WITH A DATA-CENTRIC LENS

We present DC-Check, a checklist-style framework that provides an overview of the key components of data-centric AI, summarized in Table V. Moreover, to facilitate engagement, DC-Check also serves as an actionable guide of data-centric considerations needed for reliable ML systems at a pipeline.
TABLE V
SUMMARY OF CURRENT APPROACHES WITH OPPORTUNITIES OUTLINED IN DC-CHECK. GREEN REPRESENTS IDEAS UNIQUE TO DC-CHECK.

|                  | Current | DC-Check |
|------------------|---------|----------|
| **Data**         |         |          |
| Benchmark/Highly curated datasets |         | Proactive selection/curation |
| Fixed datasets   |         | Continuous dataset curation |
| Manual data forensics |         | Automated data forensics |
| Ad hoc data preprocessing |         | Systematic data cleaning tools (AutoML/RL agents) |
| Manual dataset improvement |         | Synthetic data beyond privacy preservation |
| **Training**     |         |          |
| Performance-based model architecture search |         | Data-informed architecture selection |
| Heuristic/manual robust learning |         | Data informed robust learning |
| Domain adaptation and transfer learning |         | Improving these methods for limited data |
| Fairness and group robust methods |         | Methods to balance fairness/robustness with performance |
| Learning robust to noisy data |         | Data-centric informed usage of such methods |
| **Deployment**   |         |          |
| Limited or Low-dimensional monitoring |         | New methods for high-dimensional monitoring |
| Naive data shift remedies |         | Actionable and understandable shift remedies |
| Naive model retraining (batch) |         | Continual learning (streaming) |
| Naive dataset updates |         | Self-tuning datasets |
| Overconfident models |         | Uncertainty estimation & OOD detection |
| **Testing**      |         |          |
| Fixed data evaluation |         | Synthetic stress test-based evaluation |
| Average/population-level evaluation |         | Subset/subgroup evaluation |

Fig. 7. Overview of the interconnected flow of DC-Check across the pipeline.

The time is indeed highly pertinent for the ML community to reflect on considerations of reliable ML systems. We have passed “the making ML model work phase” to the “making real-world ML systems phase.” No longer can we simply optimize for predictive performance on benchmark tasks and assume success in the real world. This is evident as ML applications have become widespread across industries. Furthermore, the fact that high-stakes settings of healthcare and finance are beginning to implement such tools, reliability considerations should clearly be standard rather than an afterthought. Without considering the many other interconnected factors outlined by frameworks such as DC-Check, we run the risk of potentially impactful failures in the real world.

We hope that DC-Check’s actionable steps, coupled with the broader interaction of data-centric AI, will spur the research community to not only implement the data-centric tools in practice and study the checklist on real-world pipelines (and even develop a programmatic version) but also address the opportunities to realize the goal of highly performant and reliable ML systems.

APPENDIX

We present two worked examples using DC-Check which can be found in the electronic supplementary material.

1) Example 1: A dynamic pipeline for spatiotemporal fire risk prediction [135].
2) Example 2: Deep learning system for the detection of diabetic retinopathy [10], [161].

Note: The worked examples are to illustrate the usage of the DC-Check checklist and were created by the authors of
when constructing the examples. The examples are not a comment on the respective articles, but rather illustrate the usage of DC-Check on two diverse and real-world scenarios.

REFERENCES

[1] X. Zhao, “A study on E-commerce recommender system based on big data,” in Proc. IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA), Piscataway, NJ, USA: IEEE Press, 2019, pp. 222–226.

[2] P. Swarup, R. Maas, S. Garamella, S. H. Mallidi, and B. Hoffmeister, “Improving ASR confidence scores for Alexa using acoustic and hypothesis embeddings,” in Proc. Interspeech, 2019, pp. 2175–2179.

[3] F. Stahlberg, “Neural machine translation: A review,” J. Artif. Intell. Res., vol. 69, no. 69, pp. 343–418, Oct. 2020.

[4] A. Esteva et al., “Deep learning-enabled medical computer vision,” NPJ Digit. Med., vol. 4, no. 1, pp. 1–9, 2021.

[5] A. M. Alaa, D. Gurdasani, A. L. Harris, J. Rashbass, and M. van der Schaar, “Machine learning to guide the use of adjuvant therapies for breast cancer,” Nature Mach. Intell., vol. 3, no. 8, pp. 716–726, 2021.

[6] V. Moscato, A. Picarello, and G. Spertl, “A benchmark of machine learning approaches for credit score prediction,” Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 112986.

[7] R. Shwartz-Ziv and A. Armon, “Tabular data: Deep learning is not all you need,” Inf. Fusion, vol. 81, pp. 84–90, May 2022.

[8] R. Chowdhury, “Sharing learnings about our image cropping algorithm,” May 2021. [Online]. Available: https://blog.twitter.com/engineering/en_us/topics/insights/2021/sharing-learnings-about-our-image-cropping-algorithm

[9] J. Vincent, “AI camera operator repeatedly confuses bald head for soccer ball during live stream,” Nov 2020. [Online]. Available: https://www.theverge.com/tldr/2020/11/3/21547392/ai-camera-operator-football-bald-head-soccer-mistakes

[10] E. Beede et al., “A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy,” in Proc. CHI Conf. Human Factors Comput. Syst., 2020, pp. 1–12.

[11] K. Costello and M. Rimol, “Gartner identifies the top strategic technology trends for 2021,” Gartner, 2020. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2020-10-05-gartner-identifies-the-top-strategic-technology-trends-for-2021

[12] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in dataset labels,” in Proc. Adv. Neural Inf. Process. Syst., vol. 12, no. 1, pp. 56–62, 2020.

[13] S. Mäkinen, H. Skogström, E. Laaksonen, and T. Mikkonen, “Who needs MLOps: What data scientists seek to accomplish and how can MLOps help?” in Proc. Int. Conf. Mach. Learn., 2018, pp. 5689–5698.

[14] M. M. John, H. H. Olsson, and J. Bosch, “Towards MLOps: A case study for evaluating the impact of data cleaning on ml classification tasks,” in Proc. IEEE 37th Int. Conf. Data Eng., 2021, pp. 13–24.

[15] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, “Domain invariant and class discriminative feature learning for visual domain adaptation,” IEEE Trans. Image Process., vol. 27, no. 9, pp. 4260–4273, Sep. 2018.

[16] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, “ActiveClean: Interactive data cleaning for statistical modeling,” Proc. VLDB Endowment, vol. 9, no. 12, pp. 948–959, 2016.

[17] G. Csurka, “Domain adaptation for visual applications: A comprehensive survey,” 2017, arXiv:1702.05374.

[18] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, “Domain invariant and class discriminative feature learning for visual domain adaptation,” IEEE Trans. Image Process., vol. 27, no. 9, pp. 4260–4273, Sep. 2018.

[19] S. Shrivastava and A. Wong, “Auditing ImageNet: Towards a model-driven framework for annotating demographic attributes of large-scale image datasets,” 2019, arXiv:1905.01347.

[20] D. Vela, A. Sharp, R. Zhang, T. Nguyen, A. Hoang, and O. S. Pianyk, “Temporal quality degradation in AI models,” Sci. Rep., vol. 12, no. 1, pp. 1–12, 2022.

[21] F. Bayram, B. S. Ahmed, and A. Kassler, “From concept drift to model degradation: An overview on performance-aware drift detectors,” Knowl.-Based Syst., vol. 245, Jun. 2022, Art. no. 108632.

[22] W. Liang et al., “Advances, challenges and opportunities in creating data for trustworthy AI,” Nature Mach. Intell., vol. 4, no. 8, pp. 669–677, Aug. 2022.

[23] N. Polyozitis and M. Zaharia, “What can data-centric AI learn from data and ml engineering?” 2021, arXiv:2112.06439.

[24] N. Seedat, J. Crabbé, and M. van der Schaar, “Data-SUITE: Data-centric identification of in-distribution incongruous examples,” in Proc. 39th Int. Conf. Mach. Learn., PMLR, 2022, pp. 19467–19496.

[25] A. Ng, “MLOPs: From model-centric to data-centric AI,” 2021. [Online]. Available: https://www.youtube.com/watch?v=06-AZxMwHjo

[26] T. Gebru et al., “Datasheets for datasets,” Commun. ACM, vol. 64, no. 12, pp. 86–92, 2021.

[27] M. Mitchell et al., “Model cards for model reporting,” in Proc. Conf. Fairness, Accountability, Transparency, 2019, pp. 220–229.

[28] A. Jain et al., “Overview and importance of data quality for machine learning tasks,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2020, pp. 3561–3562.

[29] C. Renggli, L. Rimanic, N. M. Gürel, B. Karlas, W. Wu, and C. Zhang, “A data-quality-driven view of MLOps,” IEEE Data Eng. Bull., vol. 44, no. 1, pp. 11–24, 2021.

[30] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, “Everyone wants to do the model work, not the data work: Data cascades in high-stakes AI,” in Proc. CHI Conf. Human Factors Comput. Syst., 2021, pp. 1–15.

[31] J. W. Tukey et al., Exploratory Data Analysis, vol. 2, Reading, MA, USA: Addison-Wesley, 1977.

[32] G. J. Myatt, Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining, Hoboken, NJ, USA: Wiley, 2007.

[33] S. Morgenthaler, “Exploratory data analysis,” Wiley Interdisciplinary Rev. Comput. Statist., vol. 1, no. 1, pp. 33–44, 2009.

[34] M. L. Vigni, C. Durante, and M. Cocchi, “Exploratory data analysis,” in Data Handling in Science Technology, vol. 28, Elsevier, 2013, pp. 55–126.

[35] P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang, “CleanML: A study for evaluating the impact of data cleaning on ml classification tasks,” in Proc. IEEE 37th Int. Conf. Data Eng., 2021, pp. 13–24.

[36] S. Krishnan and E. Wu, “AlphaClean: Automatic generation of data cleaning pipelines,” 2019, arXiv:1904.11827.

[37] P. Royson and I. R. White, “Multiple imputation by chained equations (MICE): Implementation in Stata,” J. Statistical Softw., vol. 45, no. 4, pp. 1–20, 2011.

[38] D. J. Stekhoven and P. Bühlmann, “MissForest—Non-parametric missing value imputation for mixed-type data,” Bioinformatics, vol. 28, no. 1, pp. 112–118, Jan. 2012.

[39] J. Yoon, J. Jordan, and M. Schaar, “GAIN: Missing data imputation using generative adversarial nets,” in Proc. Int. Conf. Mach. Learn., PMLR, 2018, pp. 5689–5698.

[40] D. Jarrett, B. C. Cebere, T. Liu, A. Curth, and M. van der Schaar, “HyperImpute: Generalized iterative imputation with automatic model selection,” in Proc. Int. Conf. Mach. Learn., PMLR, 2022, pp. 9916–9937.

[41] G. Pleiss, T. Zhang, E. Lengenberg, and K. Q. Weinberger, “Identifying mislabeled data using the area under the margin ranking,” in Proc. Adv. Neural Inf. Process. Syst., 2020, vol. 33, pp. 17044–17056.

[42] N. Seedat, J. Crabbé, and M. van der Schaar, “Data-IQ: Characterizing subgroups with heterogeneous outcomes in tabular data,” in Proc. Adv. Neural Inf. Process. Syst., 2022, pp. 23660–23674.

[43] S. Swyamdipta et al., “Dataset cartography: Mapping and diagnosing datasets with training dynamics,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 9275–9293.
X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," in Proc. Int. Conf. Manage. Data, 2020, pp. 10842–10851.

A. Ghorbani and J. Zou, "Data Shapley: Equitable valuation of data for machine learning," in Proc. Int. Conf. Mach. Learn., PMLR, 2019, pp. 2242–2251.

C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," in Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track, 2021.

J. Jordon, J. Yoon, and M. Van Der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in Proc. Int. Conf. Learn. Representations, 2018, p. 116.

J. Yoon, L. N. Drumright, and M. Van Der Schaar, “Anonymization through data synthesis using generative adversarial networks (ADS-GAN),” IEEE J. Biomed. Health Inform., vol. 24, no. 8, pp. 2378–2388, Nov. 2020.

R. Torkzadehmahani, P. Kairouz, and B. Paten, "DP-DCGAN: Differentially private synthetic data and label generation," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2019.

L. Xu, M. Skoulidou, A. Cuesta-Infrante, and K. Veeramachaneni, “Modeling tabular data using conditional GAN,” in Proc. Adv. Neural Inf. Process. Syst., 2019, vol. 32, pp. 7335–7345.

S. Bing, A. Dittadi, S. Bauer, and P. Schwab, "Conditional generation of medical time series for extrapolation to underrepresented populations," 2022, arXiv:2201.08186.

A. Antoniou, A. Starkey, and H. Edwards, “Data augmentation generative adversarial networks,” 2017, arXiv:1711.04340.

Z. Qian, B.-C. Cebere, and M. van der Schaar, "Synthcity: Facilitating innovative use cases of synthetic data in different data modalities,” 2022, arXiv:2201.07573.

V. Shankar, R. Roelofs, H. Mania, A. Fang, B. Recht, and L. Schmidt, “Evaluating machine accuracy on ImageNet,” in Proc. Int. Conf. Mach. Learn., PMLR, 2020, pp. 8634–8644.

W. Hryniewska, P. Bombiński, P. Szatkowski, P. Tomaszewska, A. Przelaskowski, and P. Bieczek, “Checklist for responsible deep learning modeling of medical images based on COVID-19 detection studies,” Pattern Recognit., vol. 118, Oct. 2021, Art. no. 108035.

D. S. Kermany et al., “Identifying medical diagnoses and treatable diseases by image-based deep learning,” Cell, vol. 172, no. 5, pp. 1122–1131, 2018.

M. Roberts et al., “Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans,” Nature Mach. Intell., vol. 3, no. 3, pp. 191–217, 2021.

R. Jiang, S. Chiappa, T. Lattimore, A. György, and P. Kohli, “Degen-erate feedback loops in recommender systems,” in Proc. AAAI/ACM Conf. AI, Ethics Soc., 2019, pp. 383–390.

A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré, “Data programming: Creating large training sets, quickly,” in Proc. Adv. Neural Inf. Process. Syst., 2016, vol. 29, pp. 3567–3575.

A. Smola, S. Jain, P. Rajpurkar, A. Parikh, A. Ng, and M. Lungen, “Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 1500–1519.

F. Endel and H. Pringer, “Data wrangling: Making data useful again,” in Proc. 4th ACM Papers OnLine, vol. 48, no. 1, pp. 111–112, 2015.

X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, “Data cleaning: Overview and emerging challenges,” in Proc. Int. Conf. Manage. Data, 2016, pp. 2201–2206.

K. H. Tae, Y. Roh, Y. H. Oh, H. Kim, and S. E. Whang, “Data cleaning for accurate, fair, and robust models: A big data—AI integration approach,” in Proc. 3rd Int. Workshop Data Manage. End-to-End Mach. Learn., 2019, pp. 1–4.
[149] J. An and S. Cho, “Variational autoencoder based anomaly detection using reconstruction probability,” Special Lecture IE, vol. 2, no. 1, pp. 1–18, 2015.

[150] M. Sakurada and T. Yairi, “Anomaly detection using autoencoders with nonlinear dimensionality reduction,” in Proc. 2nd Workshop Mach. Learn. Sensory Data Anal. (MLSDA), 2014, pp. 4–11.

[151] J. Ren et al., “Likelihood ratios for out-of-distribution detection,” in Proc. Adv. Neural Inf. Process. Syst., 2019, vol. 32, pp. 14707–14718.

[152] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in Proc. 8th IEEE Int. Conf. Data Mining, Piscataway, NJ, USA: IEEE Press, 2008, pp. 413–422.

[153] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in Proc. Adv. Neural Inf. Process. Syst., 2018, vol. 31, pp. 7167–7177.

[154] R. K. Bellamy et al., “AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias,” IBM J. Res. Dev., vol. 63, nos. 4–5, pp. 4:1–4:15, Jul./Sep. 2019.

[155] P. Saleiro et al., “Aequitas: A bias and fairness audit toolkit,” 2018, arXiv:1811.05577.

[156] H. Baniecki, W. Kretowicz, P. Piatszek, J. Wisniewski, and P. Biecek, “Dalex: Responsible machine learning with interactive explainability and fairness in Python,” J. Mach. Learn. Res., vol. 22, no. 1, pp. 9759–9765, 2021.

[157] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Dataset Shift in Machine Learning. Cambridge, MA, USA: MIT Press, 2008.

[158] E. Breck, N. Polyzotis, S. Roy, S. Whang, and M. Zinkevich, “Data validation for machine learning,” in Proc. Mach. Learn. Syst. (MLSys), 2019, pp. 334–347.

[159] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in Proc. Adv. Neural Inf. Process. Syst., 2016, vol. 29, pp. 3323–3331.

[160] W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” in Proc. Adv. Neural Inf. Process. Syst., 2020, vol. 33, pp. 21464–21475.

[161] V. Gulshan et al., “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” JAMA, vol. 316, no. 22, pp. 2402–2410, 2016.