Algorithm Auditing: Managing the Legal, Ethical, and Technological Risks of Artificial Intelligence, Machine Learning, and Associated Algorithms

Adriano Koshiyama and Emre Kazim, University College London and Holistic AI Inc.

Philip Treleaven, University College London

Algorithms are becoming ubiquitous. However, companies are increasingly alarmed about their algorithms causing major financial or reputational damage. A new industry is envisaged: auditing and assurance of algorithms with the remit to validate artificial intelligence, machine learning, and associated algorithms.

Digital Object Identifier 10.1109/MC.2021.3067225
Date of current version: 8 April 2022

With the rise of artificial intelligence (AI) and machine learning (ML), the legal, ethical, and safety implications of their use are becoming increasingly pivotal in business and society (for example, in autonomous vehicles). We are currently entering a new phase of the "digital revolution" that might be termed *big algo* (cf. big data). We repurpose big data's 5 V's:

- *Volume*: Soon there will be billions of algorithms interacting with each other.
- *Velocity*: Algorithms are increasingly making real-time, crucial decisions with minimal human intervention.
- *Variety*: This is ubiquitous, from autonomous vehicles to medical treatment, employment, and so on.
- *Veracity*: This involves critical characteristics, such as reliability, legality, fairness, accuracy, and so on.
- *Value*: Business and government launch a proliferation of new services as sources of customer support, revenue, and cost savings, among others.

The focus in the last decade was on "data privacy"; going forward, it is "algorithm conduct."

Algorithms

An algorithm is a set of instructions designed to perform a specific task. Some perform simple tasks, like sorting a list; others try to predict the outcome of a pandemic. The former type has already been the focus of research for decades (such as in complexity and stability analysis). In this work, we are interested in the latter type, data science algorithms, which, for simplicity, we abbreviate as *algorithms*. In particular, we have AI paradigms like ML or knowledge-based systems. Although the goal of algorithm auditing is ultimately to cover the whole spectrum of algorithms and models, the focus of this article is the audit of ML algorithms.

For context, data science algorithms cover three broad domains[1]: computational statistics (for example, Monte Carlo methods), complex systems (such as agent-based systems), and AI (such as artificial neural networks).

- *Computational statistics*: These are computationally intensive statistical methods, including resampling methods (for example, bootstrap and cross-validation), Monte Carlo methods, kernel density estimation and other semiparametric and nonparametric methods, and generalized additive models.
- *Complex systems*: These systems feature a large number of interacting components whose aggregate activity is nonlinear, that is, systems featuring many interacting components whose aggregate activity is nonlinear and that typically exhibit hierarchical self-organization under selective pressures. Examples include cellular automata and multiagent systems.
- *AI algorithms*: These algorithms mimic a new form of human learning, reasoning, knowledge, and decision making:
  - o knowledge- or rule-based systems
  - o evolutionary algorithms
  - o ML.

Powerful new ML models include long short-term memory (LSTM) networks, generative adversarial networks (GANs), transfer/metalearning, and generative pretrained models (for example, GPT-3). The challenge is to interpret and manage their behavior.

Algorithm auditing

We define *algorithm auditing* as the research and practice of assessing, mitigating, and assuring an algorithm's legality, ethics, and safety. This includes the whole development process: staff training, test data sets, "circuit breaker" software, and remedial action.

Importantly, there are a number of streams feeding into the nascent field of algorithm auditing, including proposals of oversight mechanism[2]; certification ensuring software security, reliability, and functionality[3]; and the wealth of legal/information and communications technology literature on data ethics and accountability.[4] More directly, "audits" have been performed, where the audit is taken principally in terms of measuring the outcomes of a system.[5,6] By contrast, in this article we propose systemic auditing verticals and crucially do so with the various stages—from data and task setup to deployment.

Software certification may be read as the most relevant literature and practice, and it has a considerable legacy.[7] During the 1980s and 1990s, expert systems were mainly in vogue, and the main concern in relation to quality assurance was restricted to development and model.[8] We should also mention that the focus during that period was more on accuracy and computational cost. Since the 2000s, the paradigm has shifted, with most of the industrial applications of AI now relying on ML.

Current research on AI and ML includes fairness, explainability, robustness, and privacy as well as the mature topics of data ethics, management, and stewardship since we are interested in data-driven algorithms. This can be read as a maturing of the multidisciplinary field of AI governance, which has moved from numerous statements of principles toward the task of translating ethics into practice.[9,10]

As with a financial audit, an algorithm audit is required to deliver formal assurance that algorithms are legal, ethical, and safe. The stakeholders range from those working on policy and regulation, to industry practitioners and developers. We also anticipate the nature and scope of the auditing levels and framework presented will inform those interested in systems of governance and regulation/standards.

Our goal in this article is to instigate the debate on this novel area of research and practice with a robust set of areas, processes, and strategies. In a snapshot, Figure 1 outlines the dimensions and examples of activities that are part of algorithm auditing. We define them here:

- *Development*: the process of developing an algorithmic system
- *Assessment*: the process of evaluating the algorithm behavior and capacities
- *Mitigation*: the process of servicing or improving an algorithm outcome
- *Assurance*: the process of declaring that a certain system conforms to predetermined standards, practices, or regulations.

The following sections present the key components that cover algorithm audit research and practice, namely: algorithms, verticals of auditing, mitigation strategy, and assurance.

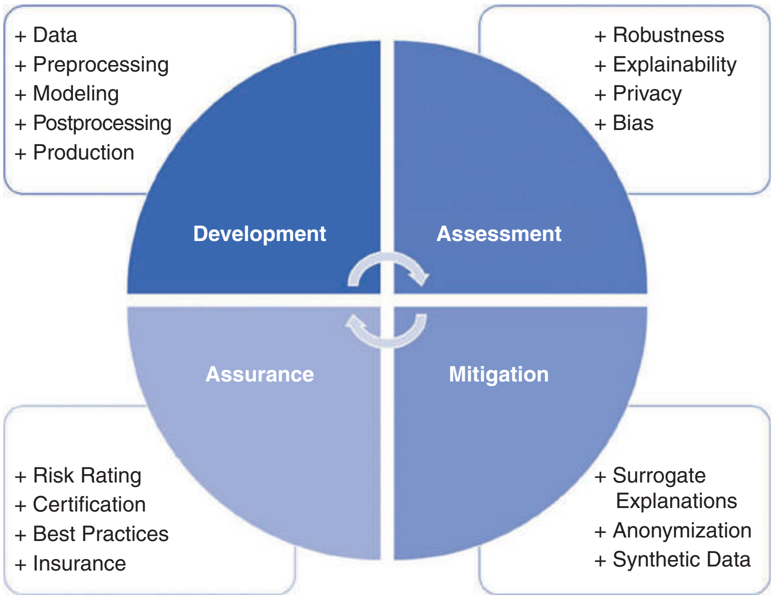

FIGURE 1. The activities of algorithm auditing.

ALGORITHMS

The key constituents of an algorithm are as follows:

- *Data*: input, output, and the simulation environment; especially important in the training of algorithms
- *Model*: objective function, formulation, parameters, and hyperparameters
- *Development*: design, building process, and infrastructure, including staff training.

For completeness, this section unpacks AI algorithms.

AI algorithms

There are broadly two classes of AI algorithms; they can be termed *static algorithms*, traditional programs that perform a fixed sequence of actions; or *dynamic algorithms*, which embody ML and can evolve. It is these latter "intelligent" algorithms that present complex technical challenges for testing and verification, which will impact and demand further regulation. These algorithms include:

- *Knowledge- or rule-based systems*: Where knowledge is explicitly represented as ontologies or IF–THEN rules rather than implicitly via code
- *Evolutionary algorithms*: A family of algorithms inspired by biological evolution, using population-based trial-and-error problem solvers with a metaheuristic or stochastic optimization character (such as genetic algorithms)

- *ML*: A type of AI program with the ability to learn without explicit programming and to change when exposed to new data; mainly comprising supervised, unsupervised, and reinforcement learning.

Additionally, deep learning, adversarial learning, transfer learning, and metalearning are new forms of learning that are ”disrupting” the prevalent supervised, unsupervised, and reinforcement learning. They are described as follows:

- *Deep learning*: This is an attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple nonlinear transformations; one example is the LSTMs.
- *Adversarial learning*: This technique is employed in the field of ML that attempts to ”fool” models through malicious input; an example is the GANs.
- *Transfer/metalearning*: These two learning paradigms are tightly connected as their main goal is to encapsulate knowledge learned across many tasks and transfer it to new, unseen ones. Knowledge transfer can help speed up training and prevent overfitting, therefore improving the obtainable final performance.

Broadly speaking, we can split the algorithm development process into governance and implementation aspects. Regarding governance, the aspects include 1) skills and training of staff, 2) accountability roles, and 3) model validation and acceptable standards for algorithm conduct. Regarding implementation, we can note five stages:

1. *Data and task setup*: collecting, storing, extracting, transforming, and loading data; ensuring that the task (regression, classification, and so on) has been well specified and designed
2. *Feature preprocessing*: enriching, transforming, and engineering the feature space
3. *In-processing/model selection*: running the whole model validation, optimization, and comparison
4. *Postprocessing and reporting*: adding thresholds and auxiliary tools to improve interpretability, presenting the results to key stakeholders, and evaluating the impact of the algorithmic system to the business
5. *Productionizing and deploying*: passing through several review processes, from IT to business, and putting in place monitoring and delivery interfaces.

Although these stages appear static and self-containing, in practice, they interact in a dynamic fashion, particularly the stages between preprocessing and postprocessing.

THE MAIN VERTICALS OF ALGORITHM AUDITING

Issues of explainability, fairness, privacy, governance, and robustness are now popular research themes among AI researchers—they are increasingly referred to as *trustworthy AI*. From an engineering point of view, we believe the most mature and impactful criteria are the following:

- *Performance and robustness*: Systems should be safe and secure, not vulnerable to tampering or compromising of the data on which they are trained.
- *Bias and discrimination*: Systems should avoid unfair treatment of individuals given their protected characteristics.
- *Interpretability and explainability*: Systems should provide decisions or suggestions that can be understood by their users and developers.
- *Algorithm privacy*: Systems should be trained following data minimization principles and should adopt privacy-enhancing techniques to mitigate personal or critical data leakage.

In Table 1, we list how each criterion interacts with the five stages of algorithm implementation. The verticals are not self-contained: they are interrelated, and there is a growing cognizance of the tradeoffs among them (see the section ”Interactions and Tradeoff Analysis”).

Performance and robustness

An algorithm’s performance and robustness are characterized by how effectively it can be deemed safe and secure. We can rate an algorithm’s robustness using four key criteria.

- *Resilience to attack and security*: Algorithmic systems, like all software systems, should be protected against vulnerabilities that can be exploited by adversaries, such as data ”poisoning,” model leakage, or the infrastructure, both software and hardware. This idea is linked with the mathematical concept of *adversarial robustness*; that is, how would the algorithm have performed in the worst-case scenario? (For example, how would the algorithm have reacted during the 2008 financial crisis?)

- *Fallback plan and general safety*: Automated systems should have safeguards that enable a fallback plan in case of problems. Also, the level of safety measures required depends on the magnitude of the risk. This notion is strongly associated with the technical concept of *formal verification*, which in broad terms means: Does the algorithm attend the problem specifications and constraints? (For example, does it respect physical laws?)
- *Accuracy*: Accuracy pertains to a system's ability to make correct judgments (such as correctly classifying information or the ability to make correct predictions, recommendations, or decisions). Accuracy as a general concept can be quantified by estimating the expected generalization performance, which means, in general, the question is asked: How well does the algorithm work? (For example, in seven out of 10 cases, does the algorithm make the right decision?)
- *Reliability and reproducibility*: A reliable automated system is one that works properly with a range of inputs and in a variety of situations, whereas reproducibility describes whether an AI experiment exhibits the same behavior when repeated under the same conditions. This idea is tied to the concept of continuous integration; that is, does the algorithm reliably reproduce its decisions?

Bias and discrimination

In AI and ML, there are multiple sources of bias leading to automated decision-making processes that become unfair.

- *Tainted training data*: Any ML system keeps the bias existing in the old data caused by human bias (for example, recruitment).
- *Skewed sample*: Future observations confirm predictions, which create a perverse feedback loop (for example, a police record).
- *Limited features*: Features may be less informative or reliably collected for minority group(s).
- *Sample size disparity*: There may be much less training data coming from the minority group than from the majority group.
- *Proxies*: Even if protected attributes are not used for training, there may be proxies for protected attributes (for example, a neighborhood).
- *Algorithm development*: The staff may not be aware or trained to assess and address biases.

To diagnose and mitigate bias in decision making, we first need to differentiate between 1) individuals, treating similar individuals similarly; and 2) groups, ensuring equality across all groups. There are multiple ways to translate these concepts mathematically. As an example, we use the SAT score as a feature for predicting success in college.

TABLE 1. Interrelations between development stage and auditing verticals.

Stage	Explainability	Robustness	Fairness	Privacy
Data and task setup	Data collection and labeling	Data accuracy	Population balance	DPIA
Feature preprocessing	Dictionary of variables	Feature engineering	Fair representations	Data minimization
Model selection	Model complexity	Model validation	Fairness constraints	Differential privacy
Postprocessing and reporting	Auxiliary tools	Adversarial testing	Bias metrics assessment	Model inversion
Productionizing and deploying	Interface and documentation	Concept drift detection and continuous integration	Real-time monitoring of bias metrics	Rate limiting and user query management

DPIA: Data Protection Impact Assessment.

- *Equality of opportunity*: The score correlates well with future success, and there is a way to use the score to correctly compare the abilities of applicants. A mathematical definition often used is the average odds difference.
- *Equality of outcome*: The SAT score may contain structural biases, so its distribution being different across groups should not be mistaken for a difference in distribution in ability. The statistical parity difference seems to be the most adopted form to represent this idea symbolically.

Interpretability and explainability

In the context of AI and ML, explainability and interpretability are often used interchangeably. *Algorithm interpretability* is about the extent to which a cause and effect can be observed and the extent to which an observer is able to predict what will happen for a given set of input or algorithm parameters. *Algorithm explainability* is the extent to which the internal mechanics of an ML (deep learning) system is explainable in human terms.

There are multiple ways to generate and provide explanations, including model-specific, model-agnostic, global, and local techniques.

- *Model specific*: A model is designed and developed in such a way that it is fully transparent and explainable by design.
- *Model agnostic*: A mathematical technique is applied to the outputs including very complex and opaque models to provide an interpretation of the decision drivers for models.
- *Global*: This facet focuses on understanding the algorithm's behavior at a high/data set/population level. Typical users are researchers and designers of algorithms since they tend to be more interested in the general insights and knowledge discovery that the model produces.
- *Local*: This method focuses on understanding the algorithm's behavior at a low/subset/individual level. Typical users of local explanations are individuals being targeted by an algorithm as well as members of the judiciary and regulators trying to make a case about potential discrimination.

Algorithm privacy

From the principles level, privacy is closely linked to the principle of prevention of "harm." It is possible to group privacy concerns into the following two key areas:

- *Privacy and data protection*: Systems must guarantee privacy and data protection throughout a system's lifecycle, and protocols governing data access should be put in place.
- *Model inferences*: The security of any system is measured with respect to the adversarial goals and capabilities that it is designed to defend against. In this sense, one needs to provide information about 1) the level of access the attacker might have ("black box" or "white box"), 2) where the attack might take place (inference or training), and 3) passive versus active attacks.

Therefore, the risk assessment of algorithm privacy can be disentangled into "data," "algorithm," and the interaction between both components. Here we outline the key methods available to assess risks coming from these elements.

- *Data*: The standard procedure to assess risks in this vertical is the Data Protection Impact Assessment (DPIA), with another vector for data poisoning, where an attacker maliciously manipulates the training data to affect the algorithm behavior.
- *Algorithm*: The key attack vector in this component is inferring model parameters to build a "knockoff" version of it. To assess vulnerability, the auditor could apply techniques that aim to extract an equivalent or near-equivalent copy or steal some functionalities of an algorithm.
- *Data–algorithm interaction*: The attack vectors in this component are inferring about members of the population or members of the training data set through interactions with the algorithm. Attacks such as statistical disclosure or model inversion are different criteria that can be applied to an algorithm to assess vulnerability.

Interactions and tradeoff analysis

Because of the interaction between verticals, a critical area of exploration is tradeoff analysis. It presents the growing evidence that in trustworthy AI there is seldom a "solution"; rather there is only management of tradeoffs. In what follows we explore two of these.

- *Explainability/interpretability versus robustness/performance*: The typical scenario is to have linear models and decision trees at the top of the explainability scale and at the bottom in terms of performance, while conversely having deep learning and support vector machines. This depiction is highly debatable in the light of data science practice since it could be that a linear model is the most accurate model, but, due to massive preprocessing performed (for example, nonlinear features), the explainability level has been drastically reduced.

- *Fairness/bias versus robustness/performance*: A standard occurrence is that, by mitigating algorithm bias, the algorithm performance is impacted. The work of an algorithm designer, therefore, is to identify acceptable boundaries of statistical bias and performance. These boundaries can be identified by liasing with business and end users and analyzing best practices, standards, or regulations commonly adopted in the field of application.

LEVELS OF ACCESS FOR AUDITING

In the literature, it is commonplace is to categorize the knowledge about the system into two extremes: ″white box″ and ″black box.″ In fact, the spectrum has more ″shades of gray,″ that is, a continuum. This additional nuance allows a richer exploration of the technologies available for assessment and mitigation as well as the right level of disclosure for a business to feel comfortable.

Hence, we can identify seven levels of system access for an auditor (Table 2), ranging from white box (full model detail disclosure) to process access (indirect observation of a system). The levels in between are set by limiting the access to the components behind the learning process (such as knowledge of the objective function, model architecture, input data, and so on).

This categorization has the following two monotonic properties:

- *Detail*: The accuracy and richness increase with the level.
- *Concealment*: The information concealed decreases with the level.

It is worth mentioning that Level 7 access allows all of the analysis of the preceding levels, simply because we have full access to the algorithm. Conversely, analysis and techniques requiring Level 7 cannot be used at Level 6 without proper assumptions. Hence, Level 7 contains all of the assessment, monitoring, and mitigation strategies of the preceding levels, with the report becoming less detailed and more inaccurate as the levels decrease.

Level 1: Process access

The auditor has no direct access to the algorithm, with its investigations and interventions occurring at the model development process. With the impossibility to perform calls at the model *f*, the auditor depends on checklists that can consist of partially qualitative and quantitative information. This level of disclosure and feedback detail is probably most appropriate for low-stakes and low-risk applications.

Level 2: Model access (black box)

The auditor has the ability to make predictive calls with the model but without having any information about the actual distributions of the input data. This level of access entails the least amount of information disclosed to the auditor since no data-sharing agreements are needed. The level of automation that can be achieved is very high since only application programming interface access is needed to perform the analysis.

Level 3: Input data access

The auditor has the capacity to make predictive calls with the model using the actual inputs used to train and validate it but cannot compare the predictions with actual outcome data. The absence of outcome information makes the problem of assessing the generalization behavior of a model difficult. Since only the predictions are available, some analysis can still be performed (for example, disparate impact) as well as property and membership inference or creating surrogate explanations.

Level 4: Outcome access (gray box)

The auditor has the capacity to make predictive calls with the model using the actual input data and to compare with outcome/output/target information. This setup is deemed black box by some authors. From a modeling perspective, a host of techniques is available to assess and operate at this level, most of them under the umbrella of ″model-agnostic″ procedures (cross-validation, Shapley values, and so on). Since there are higher levels of nonaccess, we deem this level as gray box since some information is still known to the auditor. With the available access, and based on a few assumptions, the auditor can perform concept drift analysis, investigate the accuracy of explanations, perform inversion attacks, and assess bias.

Level 5: Parameter manipulation

The auditor can recalibrate/reparameterize the model but has no information on its type or family or the incentives/objective function on which it was built. This level explicitly allows the auditor to perform stability and perturbation analysis on the model. Hence, it enables one to provide a reasonable feedback, covering in particular the areas of how stable the system is performing, its judgments, and the explanations being provided.

Level 6: Learning goal

The auditor knows most of the details encompassing the creation and purpose of the predictive system: learning procedure and task objectives, parameters, output and input data used to train and validate the model, and the access to perform predictions. From a modeling point of view, the auditor knows how to refit/relearn the model using the actual incentives/objective function on which it was trained. This level of access allows the auditor to investigate an almost accurate picture of the system without necessarily infringing on the intellectual property. The feedback has a high degree of detail.

Level 7: White-box auditing

The auditor knows all of the details encompassing the system, very much identical to those that the system developer and business user have, allowing an accurate and richer feedback. This level of access is more appropriate for internal auditors or in-house consultants since it may require nondisclosure, intellectual property sharing, data sharing, and other agreements in place. It is accurate because the whole assessment was performed using the actual

TABLE 2. The landscape of algorithm auditing.

Dimension	Level 1: Process access	Level 2: Model access $f(.)$	Level 3: Input access $f(x)$	Level 4: Outcome access $f(x), y$	Level 5: Parameter control $f_\theta(x), y$	Level 6: Learning goal $L(f_\theta(x), y)$	Level 7: White-box
Explainability	Checklist	Feature relevance, partial dependency	Surrogate explanations	Accuracy of explanations	Stability of explanations	Model complexity	Documents and specific explanations
Robustness	Checklist	Adversarial attacks	Synthetic data	Concept drift analysis	Stability analysis	Stress testing	Model selection and validation
Fairness	Checklist	Adversarial fairness	Bias in outcome	Bias in opportunity	Stability of bias metrics	Tradeoff of bias and loss metric	Model selection and development
Privacy	Checklist	Statistical disclosure	Property and membership inference	Inversion attacks	Functionality stealing	Model extraction	Model security evaluation
Information concealed	Very high	High	High	High/medium	Medium	Medium	Low
Feedback detail	Low	Medium	Medium	High/medium	High	High	Very high
Typical application	Sales forecasting	Cybersecurity	Recruitment	Credit scoring	Facial recognition	Algorithmic trading	Self-driving vehicle
Appropriate oversight	Guidelines	External auditing/ certification	External auditing	External auditing	External auditing	Internal/ external auditing	Internal Auditing

system and based on few to no assumptions; it is richer because the number of tests and recommendations that can be made range from the actual model selection to training, bias mitigation, validation, and security. It would be easier to assess mitigation strategies and provide actual information that can be more easily documented by developers.

MITIGATION STRATEGIES

Mitigation strategies are techniques employed to address issues highlighted in the auditing assessment. To some extent, they act as ”add-ons” to certain stages of model development and, hence, demand a retraining and reassessment of the model. Figure 2 establishes this feedback loop. We can highlight the following two types of mitigation procedures:

- *Human*: all procedures that involve how algorithm developers design, collaborate, reflect, and develop algorithms. These procedures can involve (re)training, impact assessment, and other tasks
- *Algorithm*: all methodologies that can be applied to improve an algorithm current outcome.

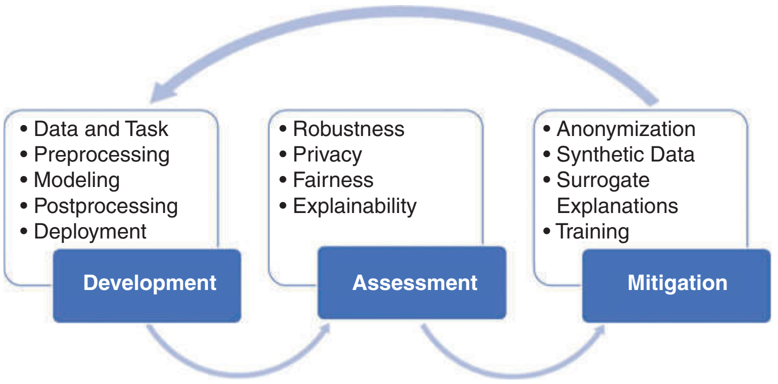

FIGURE 2. The algorithm audit process: from development, assessment, and mitigation, to redevelopment, reassessment, and reimplementing mitigation strategies.

Table 3 displays the mitigation strategies that can be employed to improve an algorithm: robustness, explainability, privacy, and fairness. Note that these strategies can be implemented jointly or in a targeted fashion; tradeoffs/interactions will be needed in this analysis.

Table 3 broadly places the strategies in the five stages of algorithm implementation upon which they can act:

- *Data and task setup*: These are simple mitigation strategies with a bigger involvement of humans, such as creating a dictionary of variables and data set sheets to improve explainability, anonymizing data to avoid leakage of personal/sensible information, labeling more information or reframing a loss function so that the model performance is improved, and searching

TABLE 3. Interrelations between implementation stage and mitigation strategies.

Stage	Explainability	Robustness	Fairness	Privacy
Data and task Setup	Dictionary of variables and data set sheets	Collecting targeted data, reframing loss function	Alternative data sources	Anonymization
Feature preprocessing	Avoiding excessive feature engineering	Feature squeezing	Synthetic data generation	Dimensionality reduction
Model selection	By-design interpretable models	Adversarial training	Counterfactual fairness	Federated learning
Postprocessing and reporting	LIME, SHAP	High confidence predictions and confidence intervals	Calibrated odds	Model inversion mitigation
Productionizing and deploying	Recourse interface	”Circuit breaking”	Monitoring panels	Rate limiting and user query management

LIME: Local interpretable model-agnostic explanations; SHAP: Shapley additive explanations.

for alternative data sources to improve bias assessment and further mitigation.

- *Feature preprocessing*: These strategies incur a mix of human intervention with the support of auxiliary algorithms, like feature squeezing to defend against adversarial attacks, avoiding excessing feature engineering to improve model interpretability, synthetic data generation to rebalance the distribution across minority groups/classes, and dimensionality reduction to address data minimization concerns.
- *Model selection* (*also known as in-processing*): Most procedures center on adding components and restrictions to the algorithm: adversarial training to increase resilience to attack and security; federated learning to avoid unnecessary data sharing; preferring by-design interpretable models like decision trees, linear regression, and so on; and using causal methods to produce ″counterfactually fair″ algorithms.
- *Postprocessing and reporting*: These are mixed human–algorithm strategies, like adding methods for model explanations, such as local interpretable model-agnostic explanations (LIME) or Shapley additive explanations (SHAP); applying procedures like calibrated odds to address any outcome/prediction bias; and adding model inversion mitigation for algorithm privacy improvement and high-confidence predictions and confidence intervals for robustness and performance.
- *Productionizing and deploying*: This involves setting up interfaces and monitors to track algorithm behavior: 1) interfaces giving users the ability to seek recourse, 2) rate limiting user queries, 3) monitoring panels to compute real-time metrics of disparate impact, and 4) circuit breaking with the algorithm trespass governance boundaries.

ASSURANCE PROCESSES

The broader outcome of an auditing process is to improve confidence or ensure trust. After assessing the system and implementing mitigation strategies, the auditing process assesses whether the system conforms to regulatory, governance, and ethical standards. Here we list the key points that embody the assurance process.

- *General and sector-specific assurance*: broad national regulations and standards (provided by agents such as the National Institute of Standards and Technology, the United Kingdom's Information Commissioner's Office, and the European Union, among others) along with sector-specific ones, such as those in financial services (such as the U.S. Securities and Exchange Commission and the U.K. Financial Conduct Authority), health (for example, the National Institutes of Health and National Health Service), and others
- *Governance*: from two aspects, namely technical assessments (for example, robustness and privacy) and impact (such as risk and compliance) assessments
- *Unknown risks*: discussing risk schemes and highlighting ″red teaming,″ used to mitigate unknown risks
- *Monitoring interfaces*: outlining risk assessments and the use of ″traffic-light″ user-friendly monitoring interfaces
- *Certification*: the numerous ways in which certification may occur, such as certification of a system or AI engineers, and so on
- *Insurance*: a subsequent service to emerge as a result of assurance maturing.

This work is a first step toward understanding the key components underlying algorithm auditing. This study is a U.K. collaboration to investigate an ″Algorithm Certification Service″ involving large industry players across different sectors, key academic institutes, and governmental bodies. In this article, we provide a list of definitions and a taxonomy since this area is a combination of research done mostly in silos, such as bias and discrimination, robustness, explainability, and privacy. Our goal is to instigate the debate in this novel area of research and practice. Translating concepts such as accountability, fairness, and transparency into engineering practice is nontrivial, with its impact perceived in design choices, algorithms used, delivery mechanisms, and built infrastructure. This demands a full integration with respect to governance structures with real-time algorithm auditing. C

ACKNOWLEDGMENTS

Adriano Koshiyama and Philip Treleaven would like to acknowledge Cisco Research Center for their research grant (2019-207109, 3696).

REFERENCES

1. A. Koshiyama, N. Firoozye, and P. Treleaven, "Algorithms in future

ABOUT THE AUTHORS

ADRIANO KOSHIYAMA is a research fellow in computer science at University College London (UCL), London, WC1E 6BT, U.K., and the cofounder of Holistic AI (www.holisticai.com), a start-up focused on providing advisory, auditing, and assurance of artificial intelligence systems. Koshiyama received a Ph.D. in computer science from UCL. He helps manage TheAlgo Conferences (thealgo.co) and is the researcher-in-house at UCL Centre of Excellence in Algorithms (Standards and Technology Lab). He is a Member of IEEE. Contact him at adriano.koshiyama.15@ucl.ac.uk.

EMRE KAZIM is a cofounder of Holistic AI, a company that provides auditing and assurance of artificial intelligence (AI) systems and a research fellow in the Computer Science Department at University College London, London, WC1E 6BT, U.K., His research interests include AI and the law, governance, assurance, and auditing of AI systems. Kazim received a Ph.D. in philosophy from Kings College London. Contact him at emre.kazim@holisticai.com.

PHILIP TRELEAVEN is director of the U.K. Centre for Financial Computing (www.financialcomputing.org) and a professor of computing at University College London, London, WC1E 6BT, U.K. His research interests are artificial intelligence applied to computational finance and regulation and "deep tech" entrepreneurship. He is a Member of IEEE. Contact him at p.treleaven@ucl.ac.uk.

capital markets," SSRN, 2020. https://ssrn.com/abstract=3527511

2. B. Shneiderman, "Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight," *Proc. Nat. Acad. Sci.*, vol. 113, no. 48, pp. 13,538–13,540, 2016, doi: 10.1073/pnas.1618211113.
3. J. Voas and K. Miller, "Software certification services: Encouraging trust and reasonable expectations," *IT Prof.*, vol. 8, no. 5, pp. 39–44, 2006, doi: 10.1109/MITP.2006.120.
4. F. Pasquale, "Data-informed duties in ai development," *Columbia Law Rev.*, vol. 119, no. 7, 2019.
5. R. E. Robertson, D. Lazer, and C. Wilson, "Auditing the personalization and composition of politically-related search engine results pages," in *Proc. 2018 World Wide Web Conf., pp.* 955–965, doi: 10.1145/3178876.3186143.
6. E. Kazim, D. M. T. Denny, and A. Koshiyama, "AI auditing and impact assessment: According to the UK information commissioner's office," *AI Ethics*, vol. 1, no. 3, pp. 301–310, 2021, doi: 10.1007/s43681-021-00039-2.
7. R. J. Ryan, "Software product quality assurance," in *Proc. AFIPS Nat. Comput. Conf.*, 1982, vol. 51, pp. 393–398, doi: 10.1145/1500774.1500823.
8. J. Rushby, "Quality measures and assurance for AI (Artificial Intelligence) software," SRI International, Menlo Park, CA, USA, Rep. no. CSL-88-7R, 1988. [Online]. Available: https://dl.acm.org/doi/10.5555/886612.
9. A. Jobin, M. Ienca, and E. Vayena, "The global landscape of AI ethics guidelines," *Nature Mach. Intell.*, vol. 1, no. 9, pp. 389–399, 2019, doi: 10.1038/s42256-019-0088-2.
10. T. Hagendorff, "The ethics of AI ethics: An evaluation of guidelines," *Minds Mach.*, vol. 30, pp. 99–120, Feb. 2020, doi: 10.1007/s11023-020-09517-8.
11. M. Brundage *et al.*, "Toward trustworthy AI development: Mechanisms for supporting verifiable claims," 2020, *arXiv:2004.07213*.
12. "Ethics guidelines for trustworthy AI," European Commission, Brussels, Belgium, [Online]. Available: https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html
13. "Ethically aligned design: Prioritizing human wellbeing with autonomous and intelligent systems," IEEE, Piscataway, NJ, USA, 2020. [Online]. Available: https://ethicsinaction.ieee.org/
14. E. Kazim and A. Koshiyama, "A high-level overview of AI ethics," SSRN, 2020. https://ssrn.com/abstract=3609292
15. B. Shneiderman, "Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered AI systems," *ACM Trans. Interactive Intell. Syst. (TiiS)*, vol. 10, no. 4, pp. 1–31, 2020, doi: 10.1145/3419764.