

Technical methods for regulatory inspection of algorithmic systems in social media platforms

A survey of auditing methods for use in
regulatory inspections of online harms

December 2021



Contents

- 3 Executive summary
- 8 How to read this report
- 10 Introduction
- 22 Auditing techniques
- 48 Recommendations
- 52 Conclusion
- 53 Methodology
- 54 Acknowledgements
- 55 Bibliography
- 60 About the Ada Lovelace Institute

Executive summary

As regulators around the world gain stronger powers to regulate online platforms – from Facebook and Twitter to TikTok – they will need robust methods to assess whether those regulatory obligations are being met.

A regulator might want to know whether a platform’s algorithms are disproportionately amplifying COVID-19 misinformation, whether actions to curb the spread of illegal content work as they claim, or if children are being recommended harmful, age-inappropriate content.

Despite widespread agreement that regulatory inspection will be a necessary part of a healthy and safe internet, there is still little agreement about what a regulatory inspection will involve, what a regulator should inspect and what methods a regulator will have at their disposal.

This report focuses specifically on methods that regulators can use during a technical audit component of a regulatory inspection. Reviewing technical auditing approaches from academia, industry and civil society, it identifies how and where they may be applicable as part of a regulatory inspection process. It details existing technical approaches for auditing online platforms, and makes suggestions for how these techniques could be used to audit content-recommendation and moderation systems.

This report also considers how these methods might be augmented by introducing new powers for state regulators that are currently out of reach of independent auditors. Motivated by UK Online Safety legislation, this report will be useful primarily to Ofcom as the designated UK online-safety regulator. However, it should also be of relevance to international regulators considering this challenge, for instance in relation to the European Digital Services Act (DSA). The report also aims to bridge independent auditing communities and policy discussions as part of the long-term ecosystem of algorithm inspection.

Our survey of algorithm audits identifies **six methods that could be applied in an online-safety context**, each of which can help regulators answer different questions about an algorithmic system, but also comes with its own limitations and challenges.

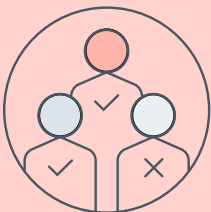
These six methods are:



1. Code audit:

Auditors have direct access to the codebase of the underlying system, or 'pseudocode' plain-English descriptions of what the code does.

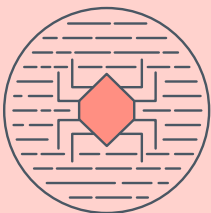
Useful for: Understanding the intentions of algorithms, and – in the case of machine learning – understanding objectives being optimised for.



2. User survey:

Auditors conduct a survey and/or perform user interviews to gather descriptive data of user experience on the platform.

Useful for: Gathering information about user experience on a platform – to paint a rough picture of the kinds of problematic behaviour that could then be further investigated.



3. Scraping audit:

Auditors collect data directly from a platform, typically by writing code to automatically click or scroll through a webpage to collect data of interest (for instance, text that users post).

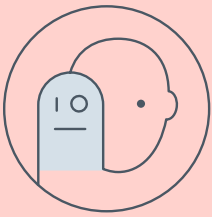
Useful for: Auditors to understand content as presented on the platform – particularly making descriptive statements (e.g. 'this proportion of search results contained this term') or comparing results for different groups or terms.



4. API audit:

Auditors access data through a programmatic interface provided by the platform that allows them to write computer programs to send and receive information to/from a platform, e.g. an API might allow a user to send a search term and get back the number of posts matching that search term.

Useful for: Giving easier programmatic access to data than a scraping audit – allowing easier automation of collection for descriptive statements or comparative work.



5. Sock-puppet audit:

Auditors use computer programs to impersonate users on the platform (these programs are called ‘sock puppets’). The data generated by the platform in response to the programmed users is recorded and analysed.

Useful for: Helping auditors to understand what a particular profile or set of profiles of users may experience on a platform.



6. Crowd-sourced audit:

Sometimes known as a ‘mystery-shopper’ audit, this method employs real users to collect information from the platform during use – either by manually reporting their experience, or through automated means like a browser extension.

Useful for: Observing what content users are experiencing on a platform, and whether different profiles of users are experiencing different content.

Based on the findings of this report, we reach the following conclusions:

1. **Technical audits form a key part of a regulatory inspection process.** The methods described in this paper can help regulators answer questions around the user experience for different kinds of users, or the prevalence of certain kinds of content online.
2. **Technical audits must be part of a wider regulatory inspection process that includes interviews and access to documentation.** Regulators will also need powers to access three kinds of evidence:¹
 - a. **Policies** – company policies and documentation that relate to the kinds of harms they are moderating for, or content they are recommending (e.g. company policy defining hate speech that guides moderation teams’ assessment practices).
 - b. **Processes** – assessment of a company’s process for identifying, removing or recommending that content (which may involve interviews with staff members).
 - c. **Outcomes** – the ability to assess the outcomes of those policies, including the behaviour of algorithmic systems that amplify or moderate content.
3. **Regulators will need explicit powers to use these methods.** Some of the methods we describe may qualify as monitoring rather than a single audit. Online-safety legislation will need to clearly carve out the ability for regulators to use these methods at their discretion.
4. **Regulators will need capacity, resources and skills to conduct these audits.** National legislation should provide regulators with resources to hire data scientists, artificial intelligence (AI) and machine learning (ML) and other technical experts to conduct these inspections. Regulators should engage with academics and civil-society organisations to help share expertise.

1 Ada Lovelace Institute and Reset. (2020). *Inspecting algorithms in social media platforms*. Available at <https://www.adalovelaceinstitute.org/wp-content/uploads/2020/11/Inspecting-algorithms-in-social-media-platforms.pdf>

5. **Policymakers must create a healthy ‘ecosystem of inspection’ by enabling civil-society and academic actors to conduct these audits.** These methods are pioneered by civil-society and academic groups, who routinely face challenges in implementing them. Policymakers must enable a marketplace of trusted, independent auditors, empower independent auditing and assessment from academic labs and civil-society organisations, and grant online-safety regulators the power to penalise platforms that actively seek to disrupt independent auditing and assessment methods or refuse to conduct such audits.

How to read this report

If you're a policymaker thinking about online harms, online platforms or regulatory approaches to algorithms

This report will help you identify potential approaches for inspecting social media platforms and hold them accountable for platform behaviours that enable or accentuate online harms. Through evidencing possible approaches, we also highlight the powers and capacity regulators will need.

- Start with the table on [page 13](#) mapping six kinds of auditing techniques, their uses and challenges.
- [Page 16](#) outlines key ways algorithms are used in social media platforms.
- See [page 48](#) for recommendations for policymakers.
- You may also be interested in our previous work *Inspecting algorithms in social media platforms*² for a more general overview of challenges, approaches and prerequisite powers.

If you're a regulator of online platforms or online harms

This report provides evidence to support thinking on the capacity, capabilities and resources you may need to fulfil a regulatory role for online platforms and harms.

- Start with the table on [page 13](#) mapping six kinds of auditing techniques, their uses and challenges.
- See the descriptions of auditing techniques from [page 22](#), which include a section on how each may be used in a regulatory context, and the challenges independent auditors have seen that regulators may be able to overcome.

2 Ada Lovelace Institute and Reset. (2020). *Inspecting algorithms in social media platforms*. Available at: <https://www.adalovelaceinstitute.org/wp-content/uploads/2020/11/Inspecting-algorithms-in-social-media-platforms.pdf>

If you're a researcher of online platforms, an independent auditor or an investigative journalist

This report translates approaches you may be familiar with in computer or social science for use as part of regulation. It supports thinking about applying theoretical approaches in practice, or existing research approaches in a policy context. There are also open questions and valuable work to be done by researchers in this space.

- Each description of an auditing technique (starting on [page 22](#)), which includes a description of their limitations and challenges to consider.
- On [page 52](#) we set out some further research questions, as well as recommendations for how researchers and independent auditors could engage with the regulatory process.

This report provides regulators and policymakers with evidence and analysis about a range of algorithm-auditing methods

Introduction

Policymakers have long been concerned with the high rates of harmful content accessible on online platforms like YouTube, Facebook and TikTok, and that concern has been exacerbated by reports of high volumes of COVID-19 misinformation spreading across these services. With the release of the Facebook Papers³ raising alarm about social media platforms' approaches to moderating or amplifying potentially harmful content, the attention of policymakers around the world has been refocused on how to regulate algorithmic systems used in social media platforms.

This report provides regulators and policymakers with evidence and analysis about a range of algorithm-auditing methods that can be applied to regulatory inspections of social media platforms, and uses the UK's forthcoming Online Safety Bill to provide contextual examples. It details existing technical approaches for auditing online platforms and makes suggestions for how these techniques could be used to audit content-recommendation and moderation systems.

In response to concerns about the prevalence of 'harmful' and illegal content online, policymakers internationally have introduced a range of regulatory packages related to online harms – the European Digital Services Act, the Irish Online Safety and Media Regulation Bill, the Canadian Online Harms consultation, and recent legislation in Australia and Germany all seek to achieve similar ends. This report is therefore of relevance to international regulators considering this challenge, and those working on technical auditing who are interested in how their work could intersect with the regulatory landscape. It aims to bridge independent auditing communities and policy discussions as part of the long-term ecosystem of algorithm inspection.

3 Wall Street Journal. (2021). *The Facebook Files: A Wall Street Journal Investigation*. Available at: <https://www.wsj.com/articles/the-facebook-files-11631713039>

Regulators will need specific powers and methods if they are to successfully inspect algorithmic systems

While the specific details of these proposals differ in their scope and remit, they all seek to provide regulators with new powers to conduct regulatory inspections of online platforms and the algorithmic systems that underpin them. A regulator responsible for assessing a platform's compliance with online-harms legislation might use one or more of these methods to investigate whether content-recommendation systems are amplifying content to users that might be illegal or harmful (e.g. if a platform's algorithms are disproportionately amplifying COVID-19 misinformation, whether actions to curb the spread of illegal content work as intended, or if children are being recommended harmful, age-inappropriate content).

Inspection powers are not a new concept for many regulators in different domains and regions, but regulators will need specific powers and methods if they are to successfully inspect algorithmic systems. The UK's draft Online Safety Bill, for example, provides Ofcom with new information-gathering powers for assessing compliance with duties of care for 'user-to-user services' like social media platforms and 'search services' like Google, to mitigate and manage risks of harm effectively.

Previous work at the Ada Lovelace Institute defined regulatory inspection of AI systems as 'a broad approach focused on an algorithmic system's compliance with regulation or norms, and requiring a number of different tools and methods'.⁴ Work from the Ada Lovelace Institute and Reset⁵ has identified the need for inspections to use various methods of evidence collection to evaluate the underlying policies, processes and outcomes of algorithmic decision-making systems. These methods may include interviews with those developing an algorithmic system, access to documentation and policies describing the system, and technical approaches for exploring a system's behaviour and effects.

This report aims to better answer the question: '*What technical methods should a regulator have at their disposal?*' by canvassing the literature of existing technical auditing approaches and identifying which methods may work for different regulatory inspection scenarios in the forthcoming online-harms regulatory regime. Academic and journalist communities have developed an extensive toolbox of technical approaches for auditing online platforms with a view to public-interest journalism and academic research, which have different goals and audiences than regulators.

4 Ada Lovelace Institute, DataKind UK. (2020). *Examining the Black Box: tools for assessing algorithmic systems*. Available at: <https://www.adalovelaceinstitute.org/report/examining-the-black-box-tools-for-assessing-algorithmic-systems/>

5 Ada Lovelace Institute, DataKind UK. (2020); Ada Lovelace Institute, Reset. (2020).

This report reviews technical auditing approaches from academia, industry and civil society, and identifies how and where they may be applicable as part of a regulatory inspection process. It also considers how these methods might be augmented by the powers of a regulator that are currently out of reach of independent auditors.

We caution that this report should not be read as suggesting that a technical audit alone is a sufficient form of regulatory inspection. A full inspection would require additional means of evidence gathering, such as interviews and documentation review, but may be bolstered by the use of these technical auditing methodologies.

What is an algorithm?

A common definition is a 'finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation'.⁶ However, there is debate about the most useful definition of an algorithm, as the term can mean different things in different contexts.⁷





As this paper focuses on algorithm audit as part of a regulatory inspection of a social media platform, in this context we consider an algorithm to be a finite sequence of well-defined computer-implementable instructions that render certain decisions on a platform (such as automatically removing certain types of content, or automatically amplifying certain types of content for users).

It is important to understand that an algorithm is **not** a social media platform – platforms may have multiple kinds of algorithms running on them at once, each seeking to perform a different function for the platform. For example, YouTube uses an algorithm to automate some of its content-moderation decisions, and uses a separate algorithm to recommend videos to users.



6 Koshiyama, A. et al. (2021) 'Towards algorithm auditing: a survey on managing legal, ethical and technological risks of AI, ML and associated algorithms.' Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3778998

7 Lum, K. and Chowdhury, R. (2021). 'What is an "algorithm"? It depends whom you ask.' *MIT Technology Review*. Available at: <https://www.technologyreview.com/2021/02/26/1020007/what-is-an-algorithm/>

Our evaluation of the literature on algorithm audits identifies six methods that could be applied in an online-harms context.⁸ Each of these methods can help regulators answer different questions about an algorithmic system, but also come with their own limitations and challenges.

Audit method	Description	Purpose	Challenges
Code audit 	Auditors have direct access to the codebase of the underlying the system, or 'pseudocode' plain-English descriptions of what the code does.	Understanding intentions of algorithms; in the case of machine learning, useful for understanding objectives are being optimised.	Codebases can be huge – individual engineers in large companies rarely understand how all parts of the platform operate. Hard to see <i>effects/outcomes</i> through code. Concerns about IP and security.
User survey 	Auditors conduct a survey and/or perform user interviews, to gather descriptive data of user experience on the platform.	Gathering information about user experience on a platform – to paint a rough picture of the types of problematic behaviour that could then be further investigated.	Vulnerable to common social science concerns with surveys – pressure to answer in a particular way, unreliable human memory and difficulty to attribute causation to findings.
Scraping audit 	Auditors collect data directly from a platform, typically by writing code to automatically click or scroll through a webpage to collect data of interest (for instance, text that users post).	Understanding content as presented on the platform – particularly making descriptive statements (e.g. 'this proportion of search results contained this term') or comparing results for different groups or terms.	Requires the development of a custom tool for each social media platform, which can be brittle as small (legitimate) changes to a website's layout can break the program.
API audit 	Auditors access data through a programmatic interface provided by the platform that allows them to write computer programs to send and receive information to/from a platform, e.g. an API might allow a user to send a search term and get back the number of posts matching that search term.	Easier programmatic access to data than a scraping audit – allowing easier automation of collection for descriptive statements or comparative work.	Publicly available APIs may not provide a regulator with the data they need. With information-gathering powers, they could compel a platform to provide access to further APIs or even a custom API, but this may require additional engineering work by platforms.

8 This taxonomy is derived from the works of Sandvig (2014) and Bandy (2021) on approaches for auditing online platforms: Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of 'Data and Discrimination: Converting Critical Concerns into Productive Inquiry'*, a preconference at the 64th Annual Meeting of the International Communication Association, p1-23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>; Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

Audit method	Description	Purpose	Challenges
Sock-puppet audit 	<p>Auditors use computer programs to impersonate users on the platform (these programs are called 'sock puppets'). The data generated by the platform in response to the programmed users is recorded and analysed.</p>	<p>Understanding what a particular user profile, or set of user profiles, may experience on a platform.</p>	<p>Sock puppets are only impersonating users – they aren't real users and so are at best a proxy for individual user activity and experience.</p>
Crowd-sourced audit 	<p>A crowd-sourced audit (sometimes known as 'mystery shopper') uses real users who collect information from the platform while they are using it⁹ – either by manually reporting experience or through automated means like a browser extension.</p>	<p>Observing what content users are experiencing on a platform and whether different profiles of users are experiencing different content.</p>	<p>Requires custom data-collection approach for each media platform being audited, often relying on web-scraping techniques; so far only demonstrated on desktop not mobile devices so may skew results or overlook mobile experiences.</p>

Understanding online platform regulation

Scenarios for regulatory inspection:

This paper's discussion of technical algorithm-auditing methods is informed by the UK's draft Online Safety Bill, currently in pre-legislative committee and due to go before Parliament in 2022. The Bill is one of the first pieces of legislation that has the potential to grant a regulator appropriate powers for inspecting algorithms in social media platforms, and therefore provides real-world legislative context to the practical application of these methods.

First proposed in 2019, and released in draft in 2021, the Online Safety Bill seeks to create a new regulatory framework for illegal and harmful online content.¹⁰ It creates duties of care for 'user-to-user services' like social media platforms, and 'search services' like Google, to mitigate and manage risks of harm effectively, and will require activities like risk assessments for illegal and harmful content.

9 Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of 'Data and Discrimination: Converting Critical Concerns into Productive Inquiry', a preconference at the 64th Annual Meeting of the International Communication Association*, p1–23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

10 Department for Digital, Culture, Media and Sport. (2021). 'Draft Online Safety Bill'. *Gov.uk*. Available at: <https://www.gov.uk/government/publications/draft-online-safety-bill>

The Bill provides Ofcom, the UK telecoms regulator whose responsibilities will be extended to online-safety regulation, with new information-gathering powers for assessing compliance with these duties. It is these information-gathering powers that could underpin regulatory inspection activities – to assess compliance with duties, a regulator needs to be able to inspect whether algorithmic systems in use by these platforms as part of their harm mitigation plans work as intended.

It is currently unclear what the standards will be for content deemed harmful, but not illegal – this is to be determined in secondary legislation. However, based on the current scope of the Bill, this report focuses on technical auditing methods that may be relevant in **two primary scenarios for regulatory inspection of social media platforms**:

1. The regulator seeks to audit the performance of **content-moderation algorithms** in use by the platform:
 - This performance is likely to be compared with the platform's risk-assessment and stated harm-mitigation moderation targets, standards or policy.
2. The regulator seeks to audit **content-recommendation algorithms** in use by the platform:
 - For example, this may be with a specific lens on terrorist content, health-related misinformation and child sexual-abuse material (content that was previously specifically highlighted in the Online Harms White Paper).¹¹

While we focus on methods applied to social media platforms ('user-to-user services' in the Online Safety Bill), these methods are also relevant to search platforms, the second focus of the Online Safety Bill. The examples we use include research conducted on search platforms, as well as social media platforms. Many social media platforms rely on internal search engines (e.g. searching for a topic or hashtag on Twitter or Instagram). Surfacing search results is similar to displaying content on social media platforms, in that it can be thought of as a question of what content to recommend in results (what content is seen most?) and what to moderate (what does not show up in results?).

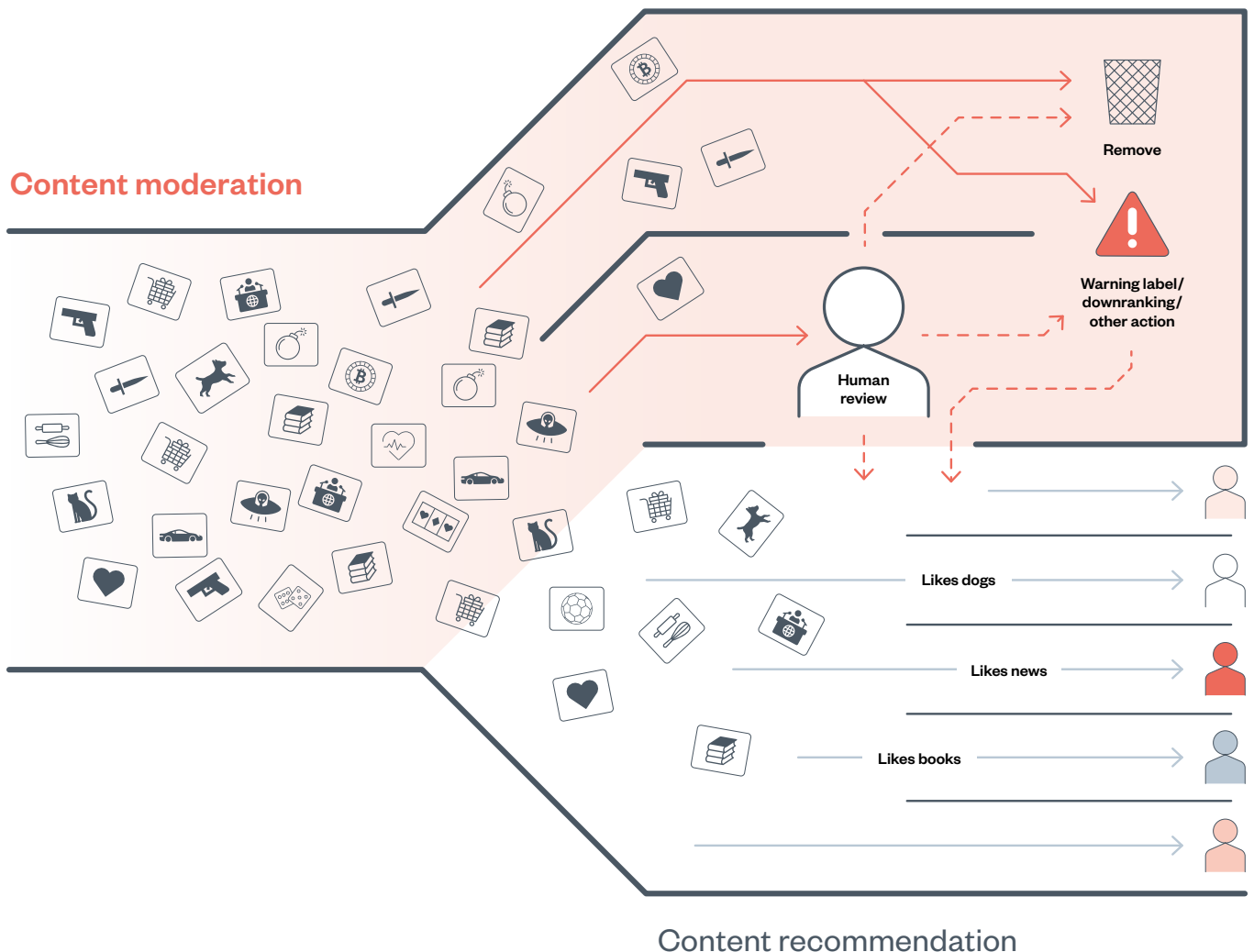
11 UK Government Department for Digital, Culture, Media & Sport and Home Office. (2019). 'Online Harms White Paper'. *Gov.uk*. Available at: <https://www.gov.uk/government/consultations/online-harms-white-paper>

The evidence in this report will be relevant for situations in which a regulator decides to perform an audit themselves and those where the regulator decides to commission an independent auditor ('skilled person' in the Online Safety Bill language) to perform the audit. In either case, a regulator will need to understand the auditing techniques that could form part of a regulatory inspection.

Two uses of algorithms in social media platforms

To illustrate how existing technical auditing methods may prove useful for a regulator that is responsible for auditing the prevalence of online harms on a platform, we will briefly describe the two common scenarios in which algorithms are used, as identified above, on social media platforms:

1. content moderation
2. content recommendation.



Many platforms and moderation service providers increasingly rely on the use of automated content moderation systems

1. Content moderation

Content moderation is the governance mechanism that structures ‘participation in a community to facilitate cooperation and prevent abuse’.¹² Every day, teams of moderators, who are sometimes hired by a platform, and at other times outsourced to third parties, flag, review, demote and remove content that has violated a platform’s terms of service. To keep up with the scale of content on these platforms, many platforms and moderation service providers increasingly rely on the use of automated content-moderation systems.

Automated content moderation is a complex sociotechnical system in which algorithms are used, typically alongside human moderators, to implement a content-moderation policy, such as identifying a piece of content to be removed from a platform. Automated content-moderation systems will vary from platform to platform, both in how they identify content and the action they take as a result.

Two primary axes along which content-moderation systems differentiate are the **type of identification**, specifically ‘**matching**’ vs ‘**prediction**’, and the **resulting action**, for instance **automatic removal vs flagging for human review**.¹³

Type of identification: How harmful content is identified in the first place by an algorithmic system

- **Matching:** Matching systems typically compare new content against a database that stores previously seen harmful content, and enable matching of new content, even if minor modifications to the previously stored content have been made.¹⁴ An example of this is the PhotoDNA system developed by Microsoft, which takes images that a human moderator has tagged as child sexual-abuse imagery and converts them into numerical ‘hash’ codes, which can then be used to automatically identify newly uploaded images to the platform that match the same hash.¹⁵

12 Grimmelman, J. (2015). ‘The Virtues of Moderation’. *Yale Journal of Law & Technology*. 7 (42). Available at: https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=2588493

13 Gorwa, R., Binns, R. and Katzenbach, C. (2020). ‘Algorithmic content moderation: technical and political challenges in the automation of platform governance’. *Big Data & Society* 7(1). Available at: <https://journals.sagepub.com/doi/full/10.1177/2053951719897945>

14 Gorwa, R., Binns, R. and Katzenbach, C. (2020).

15 Microsoft. ‘PhotoDNA’. *Microsoft.com*. Available at: <https://www.microsoft.com/en-us/photodna> [Accessed 10 November 2021].

- **Prediction:** Prediction systems typically involve training a machine-learning system on a dataset of content labelled harmful and unhelpful. This system is then deployed to classify content as it is uploaded. The canonical example of this is email spam filters, which prevent spam emails from entering your inbox. These largely operate by training a machine-learning algorithm on known instances of spam emails. The algorithm identifies common patterns and features of spam emails and can then predict the likelihood of whether a new incoming message is spam or not.

Resulting action: What happens when an automated system identifies a piece of content as harmful

- **Hard consequence:** These systems immediately block or remove the content based on the classification made, usually with potential for human review if a user appeals. The PhotoDNA system discussed above is typically used to immediately block matched child-abuse content, and a similar approach (the Shared Industry Hash Database) was used to automatically stem the spreading of the recordings of the Christchurch terrorist attack on Facebook.¹⁶
- **Soft consequence:** These systems don't immediately result in take down, and might instead be used to flag content for human review and/or affect rankings in the queue of content reviewed by human moderators. These systems are therefore best understood as human-in-the-loop systems where algorithms inform the actions of human moderators.

A regulator responsible for assessing a platform's compliance with online-harms legislation may wish to assess whether automated content-moderation algorithms are meeting their intended purpose, or if they are missing certain types of harmful content, or are moderating content that is neither illegal nor harmful.

2. Content recommendation

Another common area where an online-harms regulator may wish to conduct a technical audit relates to content-recommendation algorithms. These algorithms work in a number of ways, but their core function is to selectively surface content to users, with the intention of showing content the user will find engaging. Common examples include YouTube's video recommendation feature, TikTok's 'For You' page recommendation system, and Facebook's News Feed. There is significant evidence that these systems may recommend content that a regulator deems harmful,¹⁷ though researchers have struggled to study effectively how prevalent this material may be and how these systems may present content for different users.¹⁸

While the details of content-recommendation systems are often proprietary for technology platforms, there are a few broad categories of recommendation systems that help us understand how modern content-recommendation systems work:

- **Collaborative filtering:** At a high level, collaborative filtering computes and recommends items to users based on items liked by other users who are classified as similar. User similarity is calculated based on previous user ratings. These similarities are then used to predict ratings for items that users have not rated, and the system then recommends items that have high predicted ratings.¹⁹
 - **Matrix factorisation:** Matrix factorisation is an approach to collaborative filtering that codifies users and items into a small set of categories based on all the user ratings in a system. When Netflix recommends movies, a user may be codified by how much they like action, comedy, etc., and a movie might be codified by how much it fits into the genres of action, comedy, etc. This codified representation can then be used to guess how much a user will like a movie they haven't seen before, based on whether these codified summaries 'match'.²⁰

17 Roose, K. (2019). 'The making of a YouTube radical'. *New York Times*. Available at: <https://www.nytimes.com/interactive/2019/06/08/technology/youtube-radical.html>

18 Horta Ribeiro, M. (2021). 'Auditing radicalization pathways on YouTube'. ArXiv:1908.08313v4. Available at: <https://arxiv.org/abs/1908.08313>; https://twitter.com/random_walker/status/1211262520247439361

19 Leskovec, J., Rajaraman, A. and Ullman, J. D. (2014). 'Chapter 9: Recommendation Systems'. *Mining of Massive Datasets*. Cambridge University Press. Second edition. Available at: <https://www.cambridge.org/core/books/abs/mining-of-massive-datasets/recommendation-systems/8E2DDDAEFC644266620945386AB7DFDE>

20 Leskovec, J., Rajaraman, A. and Ullman, J. D. (2014).

- **Content-based filtering:** Content-based filtering methods recommend items based on the attributes of the item stored in the database. If the profile of items a user likes mostly consists of action films, the system will recommend other items that are tagged as action films.²¹
- **Hybrid methods:** The approaches detailed above are not mutually exclusive and can be combined in recommendation systems in particular contexts.

A regulator responsible for assessing a platform's compliance with online-harms legislation may wish to assess whether content-recommendation systems operating on a platform are amplifying or recommending content to users that may be illegal or harmful. One method for doing this is through algorithm audit.

What are algorithm audits for?

Algorithm audits can be undertaken using a variety of techniques, which we explore in detail from [page 22](#). An algorithm audit consists of probing algorithms in order to first collect data and then analyse that data for problematic patterns of interest.²² Audits are often 'empirical stud[ies]' that 'investigate a public algorithmic system for potential problematic behavior'.²³ Who defines what counts as problematic behaviour may depend on the nature of the audit. If the auditor is a third party, they may use their own definition, whereas if the audit is on behalf of a client, this definition may reflect the client's interests.

While third-party audits, performed by parties independent of the algorithm developers (often researchers, which constitute the majority of examples in this work, or investigative journalists), have motivated significant changes of technology by developers (an example is the foundational Gender Shades work that exposed intersectional identity accuracy issues in facial-analysis technology),²⁴ third-party audits often rely on public pressure and sentiment to generate accountable action.

21 Leskovec, J., Rajaraman, A. and Ullman, J. D. (2014).

22 Raji, D. and Buolamwini, J. (2019). 'Actionable auditing: investigating the impact of publicly naming biased performance results of commercial AI products'. *Association for the Advancement of Artificial Intelligence*. Available at: https://www.thetalkingmachines.com/sites/default/files/2019-02/aies-19_paper_223.pdf

23 Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

24 Buolamwini, J. and Gebru, T. (2018). 'Gender shades: intersectional accuracy disparities in commercial gender classification'. In: *Conference on Fairness, Accountability, and Transparency*, 81, p1-15. [online] New York: PLMR. Available at: <http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf>

The primary purpose of an algorithm audit is to provide public accountability for how algorithmic systems exercise power in society.²⁵ Specifically with respect to automated-content moderation, algorithm audits provide a powerful medium for making more public the discussion around what values are imbued in content-moderation systems, specifically concerning trade-offs between user safety and freedom of expression. Algorithm auditing provides the opportunity to interrogate the values and outcomes of these systems while making sure they work for society.

Content-moderation scholars have pointed out that the sheer scale of harmful content on large-scale platforms provides a good reason for the use of automated content-moderation algorithms. However, the over-reliance on automated moderation can also be a double-edged sword, effectively moving the discussion and decisions about what counts as free speech, hate speech, terrorist activity, etc. from the public eye into the private, often opaque world of corporate policies and machine-learning systems.²⁶

'Accountability' in algorithm-accountability literature is often defined with reference to the sociologist Mark Bovens, who defines it as 'a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can pose questions and pass judgement, and the actor may face consequences.'²⁷

However, the presence of an algorithm audit doesn't alone constitute a forum to which a developer or user of a technology can be held accountable, and the question of who performs the audit has significant implications for whether there is a forum with the necessary teeth for accountability.²⁸

It's important to note that algorithm auditing is only one component of the algorithm-accountability ecosystem and other factors are required to achieve true algorithm accountability.

25 Bandy, J. (2021).

26 Gorwa, R., Binns, R. and Katzenbach, C. (2020). 'Algorithmic content moderation: technical and political challenges in the automation of platform governance'. *Big Data & Society* 7(1). Available at: <https://journals.sagepub.com/doi/full/10.1177/2053951719897945>

27 Bovens, M. (2007). 'Analysing and assessing accountability: a conceptual framework'. *European Law Journal*, Vol.13, No.4

28 Moss et al. (2021). 'Assembling accountability: algorithmic impact assessment for the public interest'. *Data & Society*.

Auditing techniques²⁹



1. Code audits

Overview: In a code audit, auditors have direct access to the code of the system itself in order to perform the audit. A code audit might involve dynamic analysis (testing the code by running it on various inputs and observing the outputs) and/or manual code reviews.³⁰

What can this approach audit for?

As a code audit involves direct and transparent access to the algorithmic system, this approach in theory provides the maximal level of detail to auditors wishing to understand how a system works. However, it's important to note that many algorithmic systems consist of massive codebases, to the point that individual engineers in large companies rarely understand how all parts of the platform operate. For this reason, code audits should be looked at with a critical eye to understand what evidence can be gathered efficiently.

A useful taxonomy for the levels of access to a system an auditor might have, and what associated information can be learned at each level,³¹ specifies that each level of access is tied to a *specific model feature* that the auditor can have access to. At the lowest level of system access, the auditor has no ability to directly call or run the algorithms of interest (and this is the level of access for the majority of research surveyed in this article), and at the highest level of access, the auditor has full information on the *learning objective* (the objective the system was trained to

29 This taxonomy is derived from the works of Sandvig (2014) and Bandy (2021) on approaches for auditing online platforms: Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of "Data and Discrimination: Converting Critical Concerns into Productive Inquiry"*, a preconference at the 64th Annual Meeting of the International Communication Association, p1-23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>; Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

30 UK Competition and Markets Authority. (2021). 'Algorithms: how they can reduce competition and harm consumers'. *Gov.uk*. Available at: <https://www.gov.uk/government/consultations/algorithms-competition-and-consumer-harm-call-for-information/algorithms-how-they-can-reduce-competition-and-harm-consumers>

31 Koshiyama, A. et al. (2021). 'Towards algorithm auditing: a survey on managing legal, ethical and technological risks of AI, ML and associated algorithms'. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3778998

optimise), the ability to directly run the algorithm and access to the input data used to train the system, among other types of access.³²

This description of access points to code audits is most useful for understanding algorithm-design decisions, including the **intentions behind the system's design and objectives**. In relation to the UK Online Safety Bill, the potential for Ofcom to compel code disclosure could enable research that has previously been blocked by the proprietary nature of these systems.

However, it is unlikely that algorithmic misbehaviour is explicitly coded for (meaning that it is very unlikely that a regulatory inspection would be able to identify a problematic line of code in a company's source code), as misbehaviour often is an emergent property of the algorithmic system in operation.³³ For this reason, information gleaned from a code audit is likely to be equivalent to information that can be learned from interviews with technical and product teams responsible for algorithmic development.³⁴

A 2016 study performs a content analysis on the publicly available information on Facebook's News Feed (i.e. patents, press releases and SEC filings) in order to infer the intentions behind the algorithm driving the News Feed recommendation system, but without the higher access levels where each level of access is tied to a specific model feature, because the News Feed algorithm is proprietary.³⁵

In contrast, a 2017 study performs content analysis of the source code of open-source, mobile-news applications in order to understand the human values codified in computer programs that are performing automated content curation.³⁶ With direct access to open-source code, the researchers are able to articulate the details of how the algorithmic systems make decisions of relevance. This was possible because the mobile-news applications' source code was under an open-source licence so it was available for inspection.

32 Koshiyama, A. et al. (2021).

33 Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of 'Data and Discrimination: Converting Critical Concerns into Productive Inquiry'*, a preconference at the 64th Annual Meeting of the International Communication Association, p1-23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

34 Interviews as a source of evidence are also covered in: Ada Lovelace Institute and Reset. (2020). *Inspecting algorithms in social media platforms*. Available at: <https://www.adalovelaceinstitute.org/wp-content/uploads/2020/11/Inspecting-algorithms-in-social-media-platforms.pdf>

35 DeVito, M. A. (2016). 'From editors to algorithms: A values-based approach to understanding story selection in the facebook news feed.' *Digital Journalism*, 5:6, 753-773. Available at: <https://www.tandfonline.com/doi/abs/10.1080/21670811.2016.1178592?journalCode=rdij20>

36 Weber, M. S. and Kosterich, A. (2017). 'Coding the news: the role of computer code in filtering and distributing news'. *Digital Journalism*, 6:3, 310-329. Available at: <https://www.tandfonline.com/doi/abs/10.1080/21670811.2017.1366865?journalCode=rdij20>

What is pseudocode?

Pseudocode is a detailed, plain-English description of the steps an algorithm executes. It is the closest possible description of how an algorithm operates without using actual code. Pseudocode can include detailed specification of the type of data an algorithm processes, the processing steps it carries out, what types of classification are made, etc. In this review of code audits, the descriptions of the findings are often equivalent to pseudocode descriptions of the algorithmic systems.

How could a code audit be used in a regulatory inspection?

A regulator charged with auditing a platform's content-moderation or recommendation algorithms could find a code audit useful for identifying the **intentions** of the engineers who designed and developed the algorithmic system. The audit would be best focused at the level of the system's pseudocode (see above) – that is, a detailed, plain-English description of how the algorithm operates, which communicates what an algorithmic system does, step-by-step, without delving into the actual programming language instructions used to execute these operations.

However, a pseudocode approach is limited in its ability to verify the accuracy or reliability of the pseudocode as a representation of the code itself. Access to code itself may therefore still be helpful and important for a regulator who may wish to fact-check the assessment of the developer against an independent assessment of the code. This would require a regulator to have technical capacity (in house, or commissioned) to conduct such an evaluation.

Returning to our contextual example, instead of viewing actual source code, Ofcom could commission a platform under audit to provide pseudocode descriptions of the content-recommendation system (for instance, what features of the user determine whether a specific piece of content is relevant, and what is the prioritisation of these features?). These descriptions would inform Ofcom as to whether the design of the algorithm is aligned with the goals of online-safety legislation.

Real-world examples of code audit:

Example	Details
Analysing algorithms used for filtering and distributing news on open-source news applications ³⁷	<p>Two professors of communications, and two technical analysts fluent in the relevant programming languages, conducted a content analysis on open-source news applications to infer the editorial decisions being made by the algorithms by virtue of how they ranked articles and classified relevance to users.</p> <p>The investigators were able to analyse and describe the method by which the news curation systems collected personalisation data on users and then used this data to classify relevance of news articles before outputting final rankings.</p> <p>This study demonstrates that a code audit can help auditors understand what sources of data are prioritised by the algorithmic system. For instance, the authors found that in one system they audited, the user's Facebook likes were used as a prioritised feature for determining relevant news.</p> <p>This illustrates that much of what an auditor gleans from a code audit is a description and understanding of what the algorithmic system is doing, i.e. what design decisions were made when constructing the algorithm (for instance, what types of user data are prioritised by the recommendation system).</p>
Australia Competition Commission v. Trivago case on misleading algorithmic ranking of hotel offers ³⁸	<p>Experts in computer science conducted a manual review of the Trivago ranking algorithm of hotel offers. The experts brought in were computer-science experts capable of reviewing the algorithm and explain its workings to the court.</p> <p>The experts' findings were descriptions of the features that were ranked highest priority for determining offer price on the website.³⁹</p> <p>Here, the court and experts were focusing only on the ranking algorithm portion of the system, and the information relevant to the audit was a description of the ordering of features by priority for determining offer ranking on the website.</p>
Code audit of a DNA forensic system ⁴⁰	<p>A code audit of a DNA forensic system was performed by a team of computer scientists with input from public defenders and forensic specialists. It revealed undesirable behaviour (that could be classified as an error in the code), where an undisclosed data-dropping function could erroneously increase the number of individuals included by the system even when their DNA did not match the DNA of interest.⁴¹</p>

37 Weber, M. S. and Kosterich, A. (2017). 'Coding the news: the role of computer code in filtering and distributing news'. *Digital Journalism*, 6:3, 310-329. Available at: <https://www.tandfonline.com/doi/abs/10.1080/21670811.2017.1366865?journalCode=rdij20>

38 Australian Competition & Consumer Commission. (2020). 'Trivago misled consumers about hotel room rates'. *Gov.au*. Available at: <https://www.accc.gov.au/media-release/trivago-misled-consumers-about-hotel-room-rates>

39 UK Competition and Markets Authority. (2021). 'Algorithms: how they can reduce competition and harm consumers'. *Gov.uk*. Available at: <https://www.gov.uk/government/consultations/algorithms-competition-and-consumer-harm-call-for-information/algorithms-how-they-can-reduce-competition-and-harm-consumers>

40 Matthews, J. et al. (2019).

41 Matthews, J. et al. (2019).

Concerns and limitations of code audits:

There are several concerns and limitations that may arise if a regulator were to use code audits:

- **Code audits are complex and time consuming.** Even if given full access to code, it is not clear whether a manual code review would produce usable results that capture the kinds of harms that a regulator is looking for. Other UK authorities have noted that a code review is likely infeasible as an audit mechanism,⁴² and researchers have noted that harmful activity is usually not explicitly coded for by a system and is instead an **emergent property**. In other words, a code review would not necessarily show what kinds of material a user is actually seeing on a platform, which may limit the utility for a regulatory code review.⁴³ As the UK Competition and Markets Authority (CMA) notes, it is likely that a code audit will centrally involve discussion of pseudocode and objectives of the algorithms in question with engineers/algorithm designers, in addition to access to documentation resulting from internal audits.⁴⁴

This also depends substantially on the size of the platform to be audited. Code auditing a small, specific algorithmic subsystem (i.e. a relatively self-contained sub-component of a platform) is likely to be much more feasible than auditing a platform where the algorithm of interest lives across multiple systems and a large codebase.

- **Code audits identify problematic behaviours but not causes.** It is likely easier (and therefore to be the preferable position for a regulator) to identify the existence of problematic behaviours on a platform as opposed to the *causes* of those behaviours. A code audit would be required to identify the causes of problematic behaviours, but isn't necessary (and may not be sufficient, as problematic behaviour may arise from the combination of algorithms operating on data) to identify and name the problematic behaviours in the first place.

42 UK Competition and Markets Authority. (2021).

43 Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of 'Data and Discrimination: Converting Critical Concerns into Productive Inquiry'*, a preconference at the 64th Annual Meeting of the International Communication Association, p1-23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

44 UK Competition and Markets Authority. (2021).

- **Code audits must preserve confidentiality.** Care must be taken to ensure that disclosure is limited to particular stakeholders, so that bad actors can't use knowledge of an algorithm to 'game the system',⁴⁵ e.g. by making their content harder for systems to flag.

Challenges for independent audits that a regulator could mitigate:

There are challenges that have been surfaced within technical auditing work done in academia, journalism or civil society that a regulator's powers and capacity could mitigate, for instance:

- **Lack of precedents and confidence.** Prior academic work using code audits is very limited according to literature reviews of algorithm auditing, and this is likely to be due to lack of access to proprietary code.⁴⁶ Code audits are therefore an underexplored research area that Ofcom could open doors for. In other words, code audits shouldn't be ruled out because of a lack of precedents, since it is almost impossible currently to conceptualise how to do one and what value they might provide when there are proprietary access barriers blocking their use.

Regulatory bodies should view code audits as an exploratory research area and should enable external researchers to continue to test out these methodologies.

- There are concerns around trade-secret protections in code audits that would need to be addressed before conducting or commissioning a full code audit. We note specifically that trade-secrets protection concerns should be examined from a security angle insofar as disclosure intended for a particular entity (i.e. Ofcom) can leak or be hacked. Regarding preserving secrecy, we note that some scholars have suggested the idea of compelling disclosure using a trusted third-party to prevent full disclosure of algorithms to a regulator.^{47 48}

45 Sandvig, C. et al. (2014).

46 Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

47 Pasquale, F. (2010). 'Beyond innovation and competition: the need for qualified transparency in internet intermediaries'. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1686043

48 Sandvig, C. et al. (2014).



2. User surveys

Overview: User surveys collect direct data from users of a platform, and are then used to form a picture of user experience on the platform. Surveys require the least interaction with a technology platform as they involve gathering data about user experience through asking the users themselves. Similar to other approaches, user surveys must be sure to recruit diverse samples along axes of interest – in an online-safety context, this could involve surveying young people to understand the kinds of content a platform’s algorithm might show them.

What can this approach audit for?

Surveys are effective at gathering information about user experience on a platform. Survey data can help paint a rough picture of the kinds of problematic behaviour that should then be further investigated in an inspection.

How could user surveys be used in a regulatory inspection?

An Ofcom survey of COVID-19 misinformation polled users and determined that social media is reported as a major source of misinformation.⁴⁹ Surveys could be used to poll targeted user populations of interest,⁵⁰ for instance, children.

49 Ofcom. (2020) *Covid-19 news and information: consumption, attitudes and behaviour*. Available at: <https://www.ofcom.org.uk/research-and-data/tv-radio-and-on-demand/news-media/coronavirus-news-consumption-attitudes-behaviour>

50 Ada Lovelace Institute and Reset. (2020). *Inspecting algorithms in social media platforms*. Available at: <https://www.adalovelaceinstitute.org/wp-content/uploads/2020/11/Inspecting-algorithms-in-social-media-platforms.pdf>

Real-world examples of user surveys:

Example	Detail
Ofcom survey on experiences of COVID-19 related misinformation ⁵¹	Ofcom's 2020 survey found that respondents reported social media as one of the largest sources of COVID-19 related misinformation. Surveys like this can help Ofcom understand what problematic behaviour UK users are experiencing on platforms before launching a more thorough investigation.
Investigation into gig worker understanding of algorithmic decisions in gig-work platforms ⁵²	Researchers performed user interviews of Lyft and Uber drivers to understand how drivers made sense of algorithmic decisions on the platform concerning features such as work assignments and performance evaluation. ⁵³ Qualitative data from user interviews can motivate theories of harm that a regulatory inspection may later further investigate.

Concerns and limitations of user surveys:

- Surveys are vulnerable to a suite of concerns commonly cited in social-science literature. In particular, surveys inherently rely on reporting based on human memory and description, and 'demand bias' (pressure to answer a particular way) can drastically skew survey results.⁵⁴ Moreover, for topics that are sensitive (which may pertain to sensitive data or difficult-to-discuss issues), users may feel a similar response bias pressure to answer incorrectly.⁵⁵
- Similar to scraping studies, surveys are not experimental studies because they involve no randomisation nor experimental manipulation. This means that any given observation found from a survey study may have several competing causal theories for why such an observation occurred.⁵⁶ It is therefore difficult to causally attribute a finding of a survey to a specific characteristic of the population.

51 Ofcom. (2020).

52 As noted by Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>; Lee et al. (2015.) 'Working with Machines: The Impact of Algorithmic and Data-Driven Management on Human Workers'. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 1603-1612. Available at: <https://dl.acm.org/doi/abs/10.1145/2702123.2702548>

53 As noted by Bandy, J. (2021).

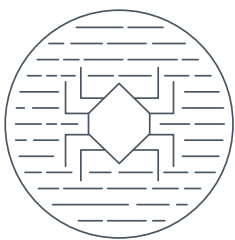
54 Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of "Data and Discrimination: Converting Critical Concerns into Productive Inquiry"*, a preconference at the 64th Annual Meeting of the International Communication Association, p1-23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

55 Sandvig, C. et al. (2014).

56 Sandvig, C. et al. (2014).

Challenges for independent audits that a regulator could mitigate:

- User survey studies require diversity along dimensions of interest (in the online-safety context, this might be age) in order to observe meaningful differences between groups.⁵⁷ Recruiting such a sample of users might require significant expense that could otherwise be prohibitive, but with fiscal capacity, Ofcom should be able to address these costs.



3. Scraping audit

Overview: A scraping audit consists of researchers collecting data directly from a platform without necessarily commissioning users to engage with the platform. This is usually done by writing code to automatically process a website's HTML/CSS (the code that the website's visual interface is written in) to collect data of interest (for instance, text that users post).

A scraping audit is a black-box method of investigating a platform, as it collects data reflecting the end-user experience without explaining how the system led to that experience, or otherwise providing any guidance as to how the system works. It allows auditors to comment on the output of how the algorithms of a platform operate together as a whole, as opposed to allowing individual inspection of sub-systems.

A core issue with scraping programs is they are often brittle – a small (legitimate) change in the layout of a website can break the logic of a scraping program. Approaches that utilise scraping methodologies must account for these changes in order to continue collecting data. For instance, the Citizen Browser team performed a crowd-sourced audit of Facebook but used scraping programs to automatically collect the Facebook News Feed data each user was shown.⁵⁸ An alternative approach would be to use public APIs to access this data (where available), as further expanded in the section on API audits on [page 35](#) – both methods provide access to public data on the platform.

⁵⁷ Sandvig, C. et al. (2014).

⁵⁸ Mattu et al. (2021). 'How we built a Facebook inspector'. *The Markup*. Available at: <https://themarkup.org/citizen-browser/2021/01/05/how-we-built-a-facebook-inspector>

What can this approach audit for?

Scraping audits can be used to collect data on a platform that can be analysed to observe statistical differences between different groups. (For example, a scraping study which used data collected from scraping to analyse correlations between the gender of a worker and their ranking on a job's platform).⁵⁹ Data obtained from scraping access is very helpful for descriptive analysis and correlational studies. In other words, scraping studies can observe patterns in the outputs of a system (but do not involve running experiments). Scraping studies therefore make descriptive statements (e.g. 'Out of a sample of X profiles on the platform that were collected via scraping, Y% displayed this characteristic'), but stop short of making statements about *causation* (i.e., that an algorithm or part of a platform caused a phenomenon to happen).

How could scraping audits be used in a regulatory inspection?

This approach could be used in one-off investigations, or to create and maintain a dataset over time that could be used for a range of inspection activities. This approach is particularly relevant for publicly available information, so might be best suited to looking at recommendations or search results: for instance, a regulator could scrape the search results for a particular term on a platform to look for prevalence of a particular type of content, or the ranking/ordering of results.

⁵⁹ Hannák et al. (2017). 'Bias in online freelance marketplaces: evidence from TaskRabbit and Fiverr'. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. p1914–1933. Available at: <https://doi.org/10.1145/2998181.2998327>

Real-world examples of scraping audits:

Example	Detail
Detailed analysis of search results ⁶⁰	<p>Scraping audits can vary significantly in scope and resourcing – from detailed examination of a single search query,⁶¹ to the construction of datasets and statistical analysis to make inferences of differential treatment.⁶²</p> <p>One of the earliest scraping studies conducted a close reading of the results returned by the Google search ‘Black girls’ to discuss the impact of search algorithms on the perception of Black women in society.⁶³</p>
Investigating bias in ranking algorithms ⁶⁴	<p>Researchers used scraping to investigate bias in ranking algorithms used by TaskRabbit and Fiverr.⁶⁵ These platforms extract demographic data, ratings and reviews, and rank of workers in search results, to build profiles of workers. Their method of analysis was to examine whether <i>correlations</i> with protected attributes (specifically gender and race) are significant with the outcomes of interest (their rating as a worker and their rank in the search results). They do this by conducting linear regressions and examining the significance of the coefficients.</p>

Concerns and limitations of scraping audits:

- This methodology is not suited to investigating causation – scraping involves collecting data not running an experiment, and scraping studies should be (and often are) careful to acknowledge this.⁶⁶
- Harmful misbehaviour of the algorithm might be observed in the combination of **algorithms and data** (e.g. due to user personalisation in content recommendation). Scraping studies are often agnostic as to whether a user is logged in: they may be conducted on public data, or from a single logged-in user account (for instance, the researcher). This is a common confusion when attempting to draw inferences about the results of personalisation algorithms from conducting scraping studies that do not attempt to model user behaviour.⁶⁷ Therefore

60 Noble, S. (2013). ‘Google Search: Hyper-Visibility as a Means of Rendering Black Women and Girls Invisible’. In *Visible Culture: An Electronic Journal for Visual Culture*. Available at: <http://ivc.lib.rochester.edu/google-search-hyper-visibility-as-a-means-of-rendering-black-women-and-girls-invisible/>

61 Noble, S. (2013).

62 Hannák et al. (2017). ‘Bias in online freelance marketplaces: evidence from TaskRabbit and Fiverr’. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. p1914–1933. Available at: <https://doi.org/10.1145/2998181.2998327>

63 Noble, S. (2013).

64 Hannák et al. (2017).

65 Hannák et al. (2017).

66 Hannák et al. (2017).

67 Narayanan, A. (2019). 29 December. Available at: https://twitter.com/random_walker/status/1211262520247439361

scraping audits alone may not be suited to investigating content-recommendation or moderation-associated harms that arise in the context of specific demographics of users (e.g. children).

- Scraping must be designed specifically for the platform being audited. Each platform has its own distinct layout of HTML and CSS code, which make up the website being displayed in the browser. The scraper must be custom built to the platform – scrapers are not multipurpose instruments and are not platform agnostic.⁶⁸ Updates to a platform may break scraping tools, jeopardising the long-term functionality of these tools.

One potential solution might be for regulators to impose requirements on platforms to make their systems more scrapable, potentially via a universal standard that all platforms must adopt. However, this solution may create other problems. First, it imposes an extremely high regulatory burden on platforms to ensure tweaks and changes to their system wouldn't break any scraping tools. Second, enabling scraping may cause personal data on a platform to become easily collectible for use in unintended or potentially harmful ways. For example, ClearviewAI trained their facial-recognition tool for law-enforcement agents on images scraped from publicly accessible social media platforms.⁶⁹

- Scraping is a black-box method that allows observation of publicly accessible parts of the platform as a whole, but doesn't necessarily allow you to identify which process of the platform caused the outcome observed in the audit. For example, if a scraping study observed a certain amount or type of harmful content on a platform, it wouldn't be able to disentangle whether the presence of that content was due to the content-moderation or content-recommendation algorithms, or a human-review process.
- Analysis of relevant control variables is limited to data that can be scraped from the platform. This could lead to drawing conclusions that suffer from omitted-variable bias, which is bias that arises in an analysis due to missing variables that have explanatory power.

68 Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of "Data and Discrimination: Converting Critical Concerns into Productive Inquiry"*, a preconference at the 64th Annual Meeting of the International Communication Association, p1–23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

69 Hill, K. (2021). 'The secretive company that might end privacy as we know it'. *The New York Times*. Available at: <https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html>

As an example, one scraping study⁷⁰ could not get access to geolocation data and therefore could not control for location. This was an issue as hiring workers based on proximity may be seen as reasonable but is correlated with race in segregated areas, thus leading to potential confounding of conclusions. It should be noted this challenge is unique to scraping and could be resolved by an API providing this information (see API audits, [page 35](#)).

- Special care and attention should be paid to how the data that is collected from a scraping audit is then processed with an eye towards not introducing bias in this stage of the process. For instance, scraping a service that uses facial data and then labelling that facial data with an off-the-shelf algorithm can introduce bias that isn't attributable to the original algorithm being audited.⁷¹

This can be a problem when the auditor wishes to audit with respect to certain characteristics not present in the data collection (most often protected characteristics). One study, for example, had to manually label worker images and profiles for gender and race.⁷²

- Once the data is collected from a scraping audit, any statistical testing is inherently tied to the domain in which the data was collected,⁷³ and so must be modelled specifically to the platform in question.

Challenges for independent audits that a regulator could mitigate:

- In some jurisdictions there have been legal challenges to scraping audits and concerns as to whether the act of web scraping violates platforms' terms and conditions. For instance, in the USA there were concerns that this may breach the Computer Fraud and Abuse Act (CFAA). However, rulings in district court have made it clear scraping, and activities to probe algorithmic systems for discrimination that breaches the terms of service are not in

70 Hannak et al. (2017).

71 UK Competition and Markets Authority. (2021). 'Algorithms: how they can reduce competition and harm consumers'. *Gov.uk*. Available at: <https://www.gov.uk/government/consultations/algorithms-competition-and-consumer-harm-call-for-information/algorithms-how-they-can-reduce-competition-and-harm-consumers>

72 Hannak et al. (2017).

73 UK Competition and Markets Authority. (2021).

violation of the CFAA,⁷⁴ and more recently the US Supreme Court ruled that violations of websites' terms of service and work to investigate online practices this way do not violate the CFAA.⁷⁵

Regulators would avoid this concern by having explicit powers to conduct this work, and could help in clarifying the rights of others to perform scraping audits and encouraging openness of platforms to these approaches.



4. API audit

Overview: A closely related type of audit to scraping audits, API audits involve interacting programmatically with an Application Programming Interface (API) instead of scraping the webpage a user sees. An API is a programmatic interface, provided by the platform, that allows external users to write computer programs to send and receive information to/from a platform (see below). The platform dictates the API that a user can interact with, which controls the information external users have access to.

A user could write a computer program to request all of the public posts made by a specific user, and the API would respond with that information in a machine-readable format. If a platform provisions an API to allow access to internal systems that are not public facing (for instance, allowing a specific internal algorithm to be queried by a user), then an API can serve to 'open up' the black box more than a scraping audit can. However, this requires the platform to provide such an API.

Unlike scraping approaches, API access does not involve processing the user-facing HTML/CSS code and instead directly exchanges the underlying data with the system. In this way, API access is a less brittle way of interacting with a system, but relies on a platform providing an API.

74 United States District Court for the District of Columbia. (2020). 'Sandvig v. Barr – memorandum opinion'. *ACLU*. [online] Available at: <https://www.aclu.org/sandvig-v-barr-memorandum-opinion>; Williams, J. (2018). *D.C. court: accessing public information is not a computer crime*. *Electronic Frontier Foundation*. [online] Available at: <https://www EFF.org/deeplinks/2018/04/dc-court-accessing-public-information-not-computer-crime> [All accessed 11.11.21].

75 *ACLU*. (2021). *Statement on Supreme Court decision removing hurdles to online civil rights testing and research*. Available at: <https://www.aclu.org/press-releases/statement-supreme-court-decision-removing-hurdles-online-civil-rights-testing-and>

What is an API?

An API defines how a computer program can send or receive information to/from another system. It can be helpful to think about APIs as the computer program equivalent to how humans interact with websites – a user of a website is provided a set of actions they can take on a website (e.g. viewing another user's posts, adding an item to a shopping cart, etc.) that is specified by the website. Similarly, a system's API specifies how an external program can perform specific actions that interact with the system, such as fetching a user's posts.

An API typically consists of **endpoints**, which are channels of communication that a program exposes that can be used to receive and send data. Access to these channels is usually restricted through the use of an **API key** which authenticates that a specific individual has access rights to interact with the API. If a user writes a program to request data from an API without a valid API key, their request will be rejected by the service. API keys also allow systems to differentiate access levels (similar to security clearances) – an API might provide certain users with a base level of access, and other users who have special permissions with an increased level of access, meaning they can perform different actions with respect to the system, for instance, fetching different kinds of data not exposed to base users.

As an example, Twitter provides an API for external developers to interact with Twitter data. The API has several different access levels, ranging from 'standard' to 'enterprise' based on the level of access, with different access levels allowing users to access different endpoints.⁷⁶ An example of an interaction a user might have with the Twitter API is requesting the data of conversations that have happened in the last week about a particular topic, based on whether the tweet contained keywords related to that topic or relevant hashtags.⁷⁷

What can this approach audit for?

As with scraping audits, data obtained in this way is suited to descriptive analysis and correlational studies focused on observing patterns in the outputs of a system. For instance, a social media platform may offer a search API for their content – if given a keyword it would return the same results as if a user searched the term on the platform. This would allow collection and analysis of results, or comparison of results of different search terms.

⁷⁶ Twitter. (n.d). *Twitter API documentation*. Available at: <https://developer.twitter.com/en/docs/twitter-api> [Accessed 11.11.2021].

⁷⁷ Twitter. (n.d).

How could an API audit be used in a regulatory inspection?

In the context of a regulatory inspection, a platform's API could be used by a regulator to write programs that request data from the platform that is deemed relevant to an inspection. To return to our contextual example, the API could be provisioned such that only UK regulator Ofcom has access (through the use of API keys provisioned for Ofcom). For Ofcom to access platform data via an API, Ofcom will require technical staff on hand to write computer programs to interact with the API, collect and store data received from the API, and process that data in the course of the inspection.

As described above, the platform determines the API provided and therefore the data that Ofcom has access to. With information-gathering powers under the Online Safety Bill, Ofcom may have the power to compel platforms to provide APIs necessary for inspection (for instance, to provide an Ofcom-only API that allows Ofcom to obtain content recommendations for a specific user). In this way, the Ofcom-only API could allow access to the internal subsystems that power the platform's content moderation and recommendation algorithms, which is a level of access not available via the standard public API. It's important to note that requiring a platform to provision a regulator-specific API would impose a regulatory burden; the platform would have to allocate engineering effort to develop these APIs for Ofcom and ensure they are safe to use.

A specific hypothetical example is that the API could provision restricted access for Ofcom (via authorised API keys) to access feed/recommended data for a sample of users where the user identity has been anonymised except for certain restricted characteristics relevant to the audit (in the online harms context, this might be age when auditing for content presented to children). The auditor could then make descriptive statements regarding the types of content that the platform surfaces to this sample of users through their feeds. The API audit is therefore able to collect data to make statements about the types and amount of content shown to users of interest, to judge whether the platform's moderation and recommendation systems are in compliance.

Real-world examples of API audits:

Example	Detail
Auditing race and gender bias in facial-recognition APIs	<p>The Gender Shades study interacts with computer vision APIs to collect classification data on how various facial-recognition platforms classify faces of different colour and gender. The study generates observational data that shows disparate impact of the algorithm as evidenced by significantly varying classification rates across different gender and race groups.⁷⁸</p> <p>The Gender Shades work was carried out by a team of computer scientists, who were responsible for data collection and running of experiments using commercial facial-recognition APIs, and a surgical dermatologist who informed benchmarking labels.</p> <p>The study was repeated a year later, finding that all target systems of the original audit had released new API versions with reduced accuracy disparities.⁷⁹</p>

Concerns and limitations of API audits:

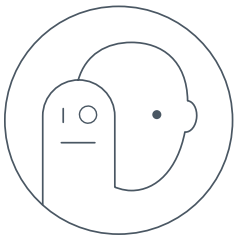
- This approach relies upon the availability of an API – this may be a public API or researcher-specific API already offered by the platform, or it may need to be custom provisioned for this purpose. Using an existing API limits the approach to what is currently made available through these APIs, which will vary from platform to platform, while the latter would require additional engineering work by platforms.
- A custom program to access the API will need to be written for each platform's API. While this is simpler than building a scraping tool, it may also require processing the data accessed via the API to make it more comparable across platforms.
- This approach relies on the API providing the same results as would be provided on the platform's website. This may not be the case for existing public APIs, and would rely on trust or scraping work to verify in the case of a custom-provided API for a regulator, unless direct access was provided to the APIs used in the website itself.

78 Buolamwini, J. and Gebru, T. (2018). 'Gender shades: intersectional accuracy disparities in commercial gender classification'. In: *Conference on Fairness, Accountability, and Transparency*, 81, p1-15. [online] New York: PLMR. Available at: <http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf>

79 Raji, I., Buolamwini, J. (2019). 'Actionable auditing: investigating the impact of publicly naming biased performance results of commercial AI products'. In: *Conference on Artificial Intelligence, Ethics, and Society (AIES)*. [online] Honolulu: ACM. Available at: <https://dl.acm.org/doi/10.1145/3306618.3314244>

Challenges for independent audits that a regulator could mitigate:

- Returning to our contextual example, using its information-gathering powers, UK regulator Ofcom could compel companies of interest to provision certain APIs. In other words, **Ofcom has the power to determine what an ideal API should look like**, as opposed to tailoring their audit methodology to the status of the provided API.



5. Sock-puppet audit

Overview: A sock-puppet audit involves using computer programs to impersonate users on the platform (these programs are called ‘sock puppets’). The data generated by the platform in response to the programmed users is recorded and stored for later analysis. This enables analysis of algorithms that involve a degree of personalisation, as programs can impersonate various demographics of interest.

What can this approach audit for?

Sock-puppet audits deploy automated programs to simulate real-life users, so they can be controlled to a finer degree than human subjects used in real-life audits.⁸⁰ Since sock-puppet audits enable manipulation of certain characteristics, they are closer to an experimental set-up than scraping audits (which are inherently more observational). Sock-puppet audits enable construction of profiles according to desired characteristics, so that the auditor can observe statistical differences in the types of data the platform surfaces to each profile along characteristics or classes of interest. Because the analysis uses profiles tailored to various characteristics of interest, this approach allows auditors to examine how personalisation algorithms vary content displayed between profiles.

One study provides a useful example of a sock-puppet audit for characterising content recommendation to users that enables researchers to draw conclusions about the personalisation elements of the algorithm.⁸¹ Specifically, the authors set up a sock-puppet program that browses the *New York Times* and collects recommendations made

80 Asplund, J. et al. (2020). ‘Auditing race and gender discrimination in online housing markets’. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1), pp. 24-35. Available at: <https://ojs.aaai.org/index.php/ICWSM/article/view/7276>

81 Chakraborty, A. and Ganguly, N. (2018). ‘Analyzing the News Coverage of Personalized Newspapers’ *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 540-54. Available at: <https://dlnext.acm.org/doi/abs/10.5555/3382225.3382338>

to the account (via a programmatic tool called Selenium).⁸² The authors simulate both a reader who randomly selects articles and readers who click articles based on articles shared by specific users on Twitter. From the collected data, they are able to quantify differences in the topic distribution shown to each sock puppet in order to judge how much the articles presented to the sock puppets vary based on their click history.

How could a sock-puppet audit be used in a regulatory inspection?

An online-safety inspection using sock puppets could involve creating sock puppets to impersonate users from different demographics (for instance, under-18 users) to use the platform and record the content recommended to them. This content could then be analysed to determine whether the amount of harmful content on the platform showed to these sensitive users is compliant with online-safety expectations. However, it should be noted that, like other approaches that collect data, such as scraping, this approach collects only a sample of data on the platform. A sock-puppet audit cannot guarantee a full picture of activity on the platform, but with a large number of sock-puppet users and content, a sample could be collected that is reasonably representative of the population of interest.

Real-world examples of sock-puppet audits:

Example	Detail
Auditing race and gender discrimination in online housing markets ⁸³	<p>Researchers used sock-puppet audits to: i) emulate demographic profiles across race and gender axes, and ii) measure differences in advertising content and number of adverts delivered to the different profiles. In particular, the puppets were trained by building browser profiles for each demographic of interest (via finding the websites that members of each demographic most commonly visited).</p> <p>To verify that the profiles had been trained successfully, the authors used statistical testing to determine whether the number and types of adverts shown in a category of advert that was representative of the profile (but independent of the features used to train the profile) was statistically significantly different from other profiles trained to be similar except in the characteristic of interest.⁸⁴</p> <p>This study was conducted by a team of computer scientists, and required significant technical expertise to build the sock-puppet profiles, train and verify them adequately, avoid bot-detection tools and collect advertising data from the sock puppets as they collected data.</p>
Understanding the influence of personalisation algorithms on news recommendation ⁸⁵	<p>Researchers used a sock-puppet audit to characterise empirically the type of news content on the <i>New York Times</i> surfaced to profiles based on their browsing characteristics. The team consisted of two computer scientists who set up the sock-puppet experiment.⁸⁶</p>
Tracking gender representation in music recommendations ⁸⁷	<p>A sock-puppet audit was used to collect data about Spotify recommendations based on bots' listening history. From analysing the collected sock-puppet data, they argue that Spotify recommendations vastly overrepresent male artists.⁸⁸</p>

Concerns and limitations of sock-puppet audits:

- As sock puppets are programs impersonating users, they aren't equivalent to real users. Sock puppets are typically programmed to emulate broad types of users, and so are at best proxies for individual user activity. For this reason crowd-sourced audits (see [page 43](#)) provide stronger evidence of real user experience.

83 Asplund, J. et al. (2020). 'Auditing race and gender discrimination in online housing markets'. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1), pp. 24-35. Available at: <https://ojs.aaai.org/index.php/ICWSM/article/view/7276>

84 Asplund, J. et al. (2020).

85 Chakraborty, A. and Ganguly, N. (2018). 'Analyzing the News Coverage of Personalized Newspapers'. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 540-54. Available at: <https://dlnext.acm.org/doi/abs/10.5555/3382225.3382338>

86 Chakraborty, A. and Ganguly, N. (2018).

87 Eriksson, M. and Johansson, A. (2017). 'Tracking gendered streams'. *Culture Unbound: Journal of Current Cultural Research*, Vol. 9 No. 2 (2017): Discovering Spotify. Available at: <https://doi.org/10.3384/cu.2000.1525.1792163>

88 Eriksson, M. and Johansson, A. (2017).

- There is a common concern that it may not be possible to analyse a platform as a sock-puppet user *without* altering the system (when the puppets are typically referred to as ‘carrier puppets’). A valid pushback to results generated from sock-puppet audits is that they reflect changes to the system caused by the sock puppets’ activities on the platform, and therefore conclusions from a sock-puppet audit may not be totally attributable to the platform itself. However, for large platforms this is likely to be an almost statistically irrelevant concern, as the number of sock puppets relevant to the total user base is quite small.⁸⁹ This concern is also only relevant where it is known that individual user activity has an effect on the platform’s interactions with other users and there is reason to believe the number of sock puppets is large enough to cause a noticeable difference.

Challenges for independent audits that a regulator could mitigate:

- A core constraint researchers face when conducting sock-puppet audits includes training, and verifying that profiles emulate the desired demographics (because this can’t be directly set as a profile characteristic, or the platform isn’t transparent about whether it is using those characteristics in its personalisation algorithms, or it isn’t clear if the platform is doing inference on those characteristics to direct adverts).⁹⁰ To improve the robustness of sock-puppet audits, Ofcom should encourage platforms to enable the simulation of profiles with particular characteristics as inferred by the algorithm.
- In the USA there were initially concerns about whether sock-puppet audits are prohibited by the Computer Fraud Abuse Act (CFAA). Automated sock-puppet accounts in particular may violate platforms’ terms of service with the use of bots. The recent Supreme and District Court rulings in favour of auditing approaches did, however, find these approaches not to be in violation of the CFAA.⁹¹ To the extent that similar concerns arise with the Computer Misuse Act in the UK, Ofcom, as a regulator with information-gathering powers, should consider supporting methods that researchers might not have the legal capacity to pursue otherwise.

89 Sandvig, C. et al. (2014). ‘Auditing algorithms: research methods for detecting discrimination on internet platforms’. In *Proceedings of “Data and Discrimination: Converting Critical Concerns into Productive Inquiry”*, a preconference at the 64th Annual Meeting of the International Communication Association, p1–23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

90 Asplund et al. (2020).

91 ACLU. (2021). *Statement on Supreme Court decision removing hurdles to online civil rights testing and research*. Available at: <https://www.aclu.org/press-releases/statement-supreme-court-decision-removing-hurdles-online-civil-rights-testing-and>



6. Crowd-sourced or collaborative audit

Overview: A crowd-sourced or collaborative audit takes the same form as a sock-puppet audit, but instead real users collect information from the platform.⁹² The users might be recruited using a sourcing firm to ensure sampling diversity across identity characteristics of interest, as in the Citizen Browser project,⁹³ or recruited via a semi-automated platform such as Amazon Mechanical Turk.⁹⁴ In both cases, the data collected is from real users using the platform and recording their experiences.

This type of study is also referred to as a ‘mystery-shopper’ audit where the mystery shoppers are recruited to use the platform and record/report the data presented to them. This methodology has precedents of use in other regulatory contexts, particularly by the CMA in investigating how consumers use digital-comparison tools when making purchasing decisions in the context of a competition investigation.⁹⁵

According to some researchers,⁹⁶ this is the most promising approach for performing audits of algorithmic systems as it avoids problems associated with other auditing mechanisms. In particular, it avoids the need to inspect source code, which is a manually intensive task demanding a large amount of expertise on the behalf of the regulator, the need to survey users (as crowd-sourced audits should automatically collect data) and terms of service breaches that scraping and/or sock-puppet audits might encounter (although these concerns are less relevant for Ofcom due to its information-gathering powers).

Crowd-sourced audits are often carried out via a browser extension that users can install, so that data that users experience is collected automatically. In order to automate the collection of user-experience data on a platform, these browser extensions typically employ scraping methods (similar to the scraping audits discussed above).

92 Sandvig, C. et al. (2014). ‘Auditing algorithms: research methods for detecting discrimination on internet platforms’. In *Proceedings of “Data and Discrimination: Converting Critical Concerns into Productive Inquiry”, a preconference at the 64th Annual Meeting of the International Communication Association*, p1–23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

93 Mattu et al. (2021). ‘How we built a Facebook feed viewer’. *The Markup*. Available at: <https://themarkup.org/show-your-work/2021/03/11/how-we-built-a-facebook-feed-viewer>

94 Sandvig, C. et al. (2014).

95 Firth, A. and Martin, G. (2017). ‘CMA digital comparison tools (DCT) mystery shopping and websweep research report market study’. *UK Competition and Markets Authority*. Available at: <https://assets.publishing.service.gov.uk/media/59c937eed915d7bd5d75ddd/gfk-mystery-shopping-and-websweep-research-report.pdf>

96 Sandvig, C. et al. (2014).

What can this approach audit for?

As a crowd-sourced audit is very similar to a sock-puppet audit, except in that it uses real users to collect information from the platform as opposed to programs, the analysis is similar. Specifically, crowd-sourced audits collect sample data that reflects users' experience on the platform and then draws conclusions from the variation in data collected based on how the user profiles vary. A major benefit of this approach is that it collects direct user-experience data, which is ultimately what an online-safety inspector is interested in.

The Citizen Browser project by *The Markup* is an example of a crowd-sourced audit that uses a browser extension to scrape user-experience data from those who have consented to use the browser.⁹⁷ The project recruited over 2,000 American users of Facebook and YouTube, and paid them to install the custom browser, which then scraped data they were exposed to on both platforms.⁹⁸ This data ultimately resulted in Split Screen, a tool built by *The Markup* to visualise news posts alongside the percentage difference between groups of interest (e.g. women vs men, Trump voters vs. Biden voters) exposed to that piece of content.⁹⁹

How could a crowd-sourced audit be used in a regulatory inspection?

A crowd-sourced audit for online safety could commission a panel of UK users to use a browser extension or custom browser similar to the Citizen Browser, to collect user data and analyse the content recommended to them. This would enable a regulator to assess whether the amounts of harmful content on the platform (that is let through content-moderation mechanisms and then recommended) is compliant with regulatory expectations. It should be noted that this methodology would require a custom approach for each platform inspected (i.e. a custom browser built for collecting Facebook user-experience data won't be compatible with collecting Twitter user-experience data). These types of audits are likely be costly, as the work required to build something like the Citizen Browser is significant.

97 Mattu et al. (2021). 'How we built a Facebook feed viewer'. *The Markup*. Available at: <https://themarkup.org/show-your-work/2021/03/11/how-we-built-a-facebook-feed-viewer>

98 Mattu et al. (2021).

99 Mattu et al. (2021).

As an alternative, a regulator could require a platform to perform an internal crowd-sourced audit and report the results to the regulator. In this case, there would not be an external browser extension collecting the data, as the platform would have direct access to user data and could record interactions the selected users had on the platform (given consent to participation). This would mitigate the need for a regulator to commission or build a data-collection tool for each platform under audit, but would require trust or assurances that the platform was reporting user-experience data accurately.

Real-world examples of crowd-sourced audits:

Example	Detail
Citizen Browser tool for auditing Facebook and YouTube content and recommendations ¹⁰⁰	<p><i>The Markup's</i> Citizen Browser project was built by a team of journalists and software developers. The team commissioned an external firm to hire a panel of American users, developed a desktop browser with scraping code and performed statistical analysis on the content variations between the groups of interest.</p> <p>It should be noted that the browser's scraping code had to be custom built to the Facebook user interface, which was evolving during the course of the Citizen Browser project, meaning the scraping code was undergoing constant updates. There have also been concerns that the platform has made changes to the HTML code behind the user interface that make it more difficult for the scraping code to work.¹⁰¹</p> <p>This reflects a primary difficulty outlined earlier of approaches that use scraping – that they are fragile to user interface changes made by the platform.</p>
Investigating price discrimination in online platforms ¹⁰²	<p>Researchers conducted a crowd-sourced audit to study price discrimination in online platforms. To do so, they used a browser extension to collect (with consent) data from 340 users about prices they were shown while shopping for products. Cross referencing the user browsing data revealed variations in price for the same products, allowing the researchers to identify online stores that were 'price discriminating' against consumers.¹⁰³</p> <p>The research team was a team of computer scientists, one of whom built the browser extension that collected the relevant data.</p>

100 Mattu et al. (2021).

101 Faife, C. (2021). 'Facebook rolls out news feed change that blocks watchdogs from gathering data'. *The Markup*. Available at: <https://themarkup.org/citizen-browser/2021/09/21/facebook-rolls-out-news-feed-change-that-blocks-watchdogs-from-gathering-data>

102 Mikians, J. et al. (2013). 'Crowd-assisted search for price discrimination in e-commerce: first results'. *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, p1–6. <https://doi.org/10.1145/2535372.2535415>

103 Mikians, J. et al. (2013).

Concerns and limitations of crowd-sourced audits:

- Crowd-sourced audit tools are typically built using browser extensions or desktop browsers (the Citizen Browser is an example of a desktop browser). Sample data collected will skew towards users of platforms who access them via desktops, as opposed to mobile users. To the extent that desktop users skew along axes such as race, gender and socioeconomic status, the results will be accordingly skewed.
- Similar to approaches that rely on observing the outputs of the platform (every approach that isn't a code audit), crowd-sourced audits do not reverse engineer the recommendation algorithms of the platform. This means an auditor can't claim the recommendation algorithm caused the statistical differences in content types between user groups based on the characteristics of the users. This is specifically noted by *The Markup's* limitations of the Citizen Browser project.¹⁰⁴

Similar to scraping audits, omitted-variable bias can be an issue of any crowd-sourced audit, as there may be features that the platform uses to drive recommendations that are not visible to the auditors. This results in missing variables that might be causing the differences observed. Acknowledging this, *The Markup* notes that 'our observations should not be taken as proof of Facebook's choosing to target specific content at specific demographic groups. There are many factors that influence any given person's feed that we do not account for, including users' friends and social networks'.¹⁰⁵

- As crowd-sourced audit tools often contain scraping code to automatically collect user data, they suffer from the same obstacles as scraping, being easily breakable due to changes in the platform's user interface (i.e. its HTML and CSS).

The Citizen Browser project had to write custom scraping code for multiple Facebook user interfaces because scraping code must be custom built to a platform's HTML and CSS interface.¹⁰⁶

104 Mattu et al. (2021). 'How we built a Facebook inspector'. *The Markup*.

Available at: <https://themarkup.org/citizen-browser/2021/01/05/how-we-built-a-facebook-inspector>

105 Mattu et al. (2021).

106 Mattu et al. (2021).

- Any auditor will have to ensure that collection of user data from a crowd-sourced audit is compliant with data-protection rules. This has been a consideration in independently conducted audit work – for instance, *The Markup*'s Citizen Browser project specifically redacted user data in order to address privacy concerns, and having been first used in the USA, has since undertaken work in a bid to comply with other data-protection regimes and is now operating in Germany.¹⁰⁷

Challenges for independent audits that a regulator could mitigate:

- A major concern for researchers running a crowd-sourced or collaborative audit is the financial cost required to recruit a panel of users that is representative enough along the diverse users' axes of identity, so that a regulator like Ofcom can investigate the algorithm's treatment of those users. For content recommendation and content moderation focused on children's safety online, it will be relevant for Ofcom to keep in mind the representativeness of the sample of users for which it analyses data. With regulatory remit and fiscal capacity, Ofcom should be able to better address the costs of acquiring a panel of users necessary for the audit.

107 Angwin, J. (2021). 'Bringing the Citizen Browser Project to Germany'. *The Markup*.

Available at: <https://www.getrevue.co/profile/themarkup/issues/bringing-the-citizen-browser-project-to-germany-700391>

Recommendations

For policymakers thinking about online harms, online platforms or regulatory approaches to algorithms

The technical approaches for regulatory inspection outlined rely on regulators having sufficient powers and capacity to conduct them. As policymakers look to give regulators new powers with respect to online platforms – from recent legislation in Australia and Germany, to draft legislation in the UK, Canada and EU, and forthcoming legislation elsewhere – we recommend they consider the following:

1. **Articulating explicit powers for regulators to undertake regulatory inspections of online platforms when they deem appropriate.** This may include powers that enable regulators to access documentation about a product, conduct interviews with staff, and gather information about an algorithmic system's outcomes and the policies that the algorithm is operating under (for example, relevant hate-speech policies as defined by the platform).
2. **Articulating explicit powers that enable regulators, as part of an inspection, to perform technical audits, assessments and monitoring of platform behaviour, including algorithmic behaviour, whenever they deem appropriate.** Some of the methods described in this report may be possible for regulators to undertake today, but others may require new powers to monitor the behaviour of a platform or algorithmic system over time.
3. **Creating a healthy 'ecosystem of inspection' by:**
 - a. **Enabling a marketplace of independent auditors that platforms can turn to.** There are already some for-profit auditing firms that specialise in delivering algorithm audits of an algorithmic system's behaviour.¹⁰⁸ Online-safety legislation could help foster a new marketplace of independent auditors by granting regulators the power to mandate a platform to undertake an independent

108 Notable examples include ORCAA and Arthur.ai.

audit from a third-party agency that the regulator has vetted and approved. This marketplace is a new way to create an economic opportunity for national governments. The global AI-governance products and services market is estimated to be worth \$402 million by 2026.¹⁰⁹

- b. Empowering independent auditing and assessment from academic labs and civil-society organisations.** Online-safety bills must recognise that regulators sit within, and will rely on, a wider ecosystem of inspection in which civil-society organisations and academics are empowered to provide independent audits and assessments of platform behaviour. Many of the auditing methods we describe above fail because platforms do not provide relevant information or access to the data an auditor needs to perform these assessments. Online-safety legislation could address this by granting regulators the power to compel platforms to provide certain data or APIs to third-party auditors who can undertake their own independent audits.
 - c. Granting regulators the power to penalise platforms that actively seek to disrupt independent auditing and assessment methods or refuse to conduct such audits.** Many independent auditors from civil-society organisations and academic labs described their relationship with social media firms as one in which platforms treat them as adversaries rather than partners. In many cases, online platforms have actively disrupted efforts to run these audits. To mitigate this threat to accountability, policymakers should empower regulators with the ability to penalise or fine platforms that take steps to actively disrupt independent auditing capabilities.
- 4. Providing regulators with the capacity, resources and skills to conduct these audits.** National legislation should provide regulators with resources to hire data scientists, AI and ML experts and other technical experts to conduct these inspections. Regulators should engage with academics and civil-society organisations to help build capacity and share expertise.

109 StrategyR. (2021). *Global Artificial Intelligence (AI) Governance Market to Reach \$402 Million by 2026*. Available at: at <https://www.strategyr.com/market-report-artificial-intelligence-ai-governance-forecasts-global-industry-analysts-inc.asp>

For researchers of online platforms, independent auditors and investigative journalists

This report is primarily aimed towards answering the question: *‘What do existing technical algorithm-auditing methods look like, and how can they be used in the context of regulation?’* This leaves many important accountability questions unanswered that future work will need to address.

Moreover, because auditing algorithms and platforms in the context of online safety is nascent (and made relevant by the UK Online Safety Bill and the EU Digital Services Act), there is limited research that audits algorithms specifically in the context of online safety. As a result, this report reviews existing auditing approaches within the domains where they were originally performed, and suggests ways for these approaches to be used for online safety-related auditing. Further work is needed to fully answer future-facing questions, which will include:

1. How often should audits be performed? Should they be an internal, continuous process or viewed as one-off occasions triggered by external events? How do actors, both internal and external, verify and/or reproduce audit findings?
2. How can these accountability methods be purposed for internal auditing efforts, and what new challenges and opportunities are raised by platforms auditing themselves?
3. To what extent are algorithm audits for regulatory inspection constrained by data-protection regulation, and what will be required for algorithm audits to be compliant with the EU GDPR and any future divergent UK data-protection regime?
4. To what extent do trade-secret protections interfere with the capacity for algorithm audit, particularly in the context of code disclosure when such a code audit is deemed useful?
5. Who should perform the audit? Performing technical audits requires significant technical capacity, so regulators should invest in internal capacity, or commission external auditors to do so. Whether the regulator themselves or a third-party private auditor conducts the audit is likely to have a large impact on the answers to the two questions above.

Finally, the approaches in this work leave many important techniques for interrogating algorithmic systems unaddressed. Work that doesn't neatly fall into the algorithm-auditing techniques surveyed in this paper include research methods like development histories and case studies, among others.¹¹⁰

Development histories can help shed light on intentions behind the design of the algorithmic system, providing helpful context for regulators when considering what behaviours to audit for or what systems to audit (such as the history behind the development of predictive tools for child services in New Zealand,¹¹¹ or the values optimised for in the design of the Facebook News Feed algorithm).¹¹²

Case studies, which are typically in-depth explorations of single-instance or small-scale interactions with a platform or algorithm, can highlight problematic behaviours in sociotechnical systems to motivate later, large-scale audits.

110 Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

111 Gillingham, P. (2015). 'Predictive risk modelling to prevent child maltreatment and other adverse outcomes for service users: inside the 'black box' of machine learning'. *The British Journal of Social Work*, Volume 46, Issue 4, p1044–1058. Available at: <https://academic.oup.com/bjsw/article-abstract/46/4/1044/2472679>; Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

112 DeVito, M. A. (2016). "From editors to algorithms: a values-based approach to understanding story selection in the facebook news feed." *Digital Journalism*, 5:6, 753-773. Available at: <https://www.tandfonline.com/doi/abs/10.1080/21670811.2016.1178592?journalCode=rdij20>; Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

This paper has outlined six technical auditing methods that regulators may consider implementing as part of a regulatory inspection of a platform

Conclusion

Regulatory agencies across Europe and the UK are likely to become increasingly empowered to hold technology firms accountable for the harms their services may cause or enable. Legislative proposals like the EU's Digital Services Act and the UK's Online Safety Bill will enable national regulators to inspect platforms in a variety of ways.

This paper has outlined six technical auditing methods that regulators may consider implementing as part of a regulatory inspection of a platform. Each method seeks to answer a slightly different question for regulators – code audits, for example, can help answer questions about a developer's intentions when creating an algorithmic system, while crowd-sourced audits can help regulators understand the experience of particular users on a platform.

Each method comes with challenges and limitations, but each may serve a particular purpose for regulators seeking to inspect platforms for the prevalence of harmful behaviour. We have also highlighted opportunities and benefits for regulators in enabling third-party auditors, including those from the research community, investigative journalism and civil society who have developed these methods, to more easily continue this work.

Our intention is that this report will contribute to the emerging discourse around regulatory inspection and algorithm audits, and help regulators developing the capacity to address harmful online behaviour to develop a robust, flexible and effective accountability toolkit.

Methodology

This report surveys auditing techniques already used in academic research, investigative journalism, civil society and industry, and considers their application to a regulatory context.

The report has been developed primarily through a desk-based review and synthesis of technical documentation, grey and academic literature on auditing methodologies. It is also informed by policy analysis of white papers and draft legislation related to online harms, primarily in a UK and European context. This analysis has been limited to legislation drafted in English, and we would welcome further work considering wider linguistic, geographic and political contexts.

This research does not examine methods around auditing algorithmic systems for biases, which are likely to fall outside the scope of the UK Online Safety Bill, and which have a more extensive literature around methods.¹¹³ While it touches on data-protection considerations of these methods, we do not address these in depth and identify this as an area for further work.

113 Raji, D. and Buolamwini, J. (2019). 'Actionable auditing: investigating the impact of publicly naming biased performance results of commercial AI products'. *Association for the Advancement of Artificial Intelligence*. Available at: https://www.thetalkingmachines.com/sites/default/files/2019-02/aies-19_paper_223.pdf

Acknowledgements

We would like to thank the following colleagues for taking time to review a draft of this paper or offering their expertise and feedback:

- Aparna Surendra, AWO
- Eric Kind, AWO
- Jack Bandy, Computational Journalism Lab, Northwestern University
- Rumman Chowdhury, ML Ethics, Transparency, and Accountability, Twitter.

This report was lead authored by Aneesh Pappu, with substantive contributions from Jenny Brennan, Andrew Strait, Imogen Parker and Elliot Jones.

Bibliography

ACLU. (2021). 'Statement on Supreme Court decision removing hurdles to online civil rights testing and research'. Available at: <https://www.aclu.org/press-releases/statement-supreme-court-decision-removing-hurdles-online-civil-rights-testing-and>

Ada Lovelace Institute, DataKind UK. (2020). *Examining the Black Box: tools for assessing algorithmic systems*. Available at: <https://www.adalovelaceinstitute.org/report/examining-the-black-box-tools-for-assessing-algorithmic-systems/>

Ada Lovelace Institute and Reset. (2020). *Inspecting algorithms in social media platforms*. Available at: <https://www.adalovelaceinstitute.org/wp-content/uploads/2020/11/Inspecting-algorithms-in-social-media-platforms.pdf>

Asplund, J. et al. (2020). 'Auditing race and gender discrimination in online housing markets'. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1), pp. 24-35. Available at: <https://ojs.aaai.org/index.php/ICWSM/article/view/7276>

Bandy, J. (2021). 'Problematic machine behavior: a systematic literature review of algorithm audits'. *Proceedings of the ACM on Human-Computer Interaction*. Volume 5, Issue CSCW1. Available at: <https://doi.org/10.1145/3449148>

Bovens, M. (2007). 'Analysing and assessing accountability: a conceptual framework'. *European Law Journal*, Vol. 13, No. 4, pp. 447-468. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=992006

Buolamwini, J. and Gebru, T. (2018). 'Gender shades: intersectional accuracy disparities in commercial gender classification'. In: *Conference on Fairness, Accountability, and Transparency*, 81, p1-15. [online] New York: PLMR. Available at: <http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf>

Chakraborty, A. and Ganguly, N. (2018). 'Analyzing the News Coverage of Personalized Newspapers.' 2018 IEEE/ACM International Conference on Advances in *Social Networks Analysis and Mining (ASONAM)*, pp. 540-54. Available at: <https://dlnext.acm.org/doi/abs/10.5555/3382225.3382338>

Department for Digital, Culture, Media and Sport. (2021). 'Draft Online Safety Bill'. *Gov.uk*. Available at: <https://www.gov.uk/government/publications/draft-online-safety-bill>

DeVito, M. A. (2016). 'From editors to algorithms: A values-based approach to understanding story selection in the facebook news feed.' *Digital Journalism*, 5:6, 753-773. Available at: <https://www.tandfonline.com/doi/abs/10.1080/21670811.2016.1178592?journalCode=rdij20>

Eriksson, M. and Johansson, A. (2017). 'Tracking gendered streams'. *Culture Unbound: Journal of Current Cultural Research*, Vol. 9 No. 2 (2017): Discovering Spotify. Available at: <https://doi.org/10.3384/cu.2000.1525.1792163>

Faife, C. (2021). 'Facebook rolls out news feed change that blocks watchdogs from gathering data'. *The Markup*. Available at: <https://themarkup.org/citizen-browser/2021/09/21/facebook-rolls-out-news-feed-change-that-blocks-watchdogs-from-gathering-data>

Firth, A. and Martin, G. (2017). 'CMA digital comparison tools (DCT) mystery shopping and websweep research report market study'. *UK Competition and Markets Authority*. Available at: <https://assets.publishing.service.gov.uk/media/59c937eed915d7bd5d75ddd/gfk-mystery-shopping-and-websweep-research-report.pdf>

Gillingham, P. (2015). 'Predictive risk modelling to prevent child maltreatment and other adverse outcomes for service users: inside the 'black box' of machine learning'. *The British Journal of Social Work*, Volume 46, Issue 4, p1044-1058. Available at: <https://academic.oup.com/bjsw/article-abstract/46/4/1044/2472679>

Gorwa, R., Binns, R. and Katzenbach, C. (2020). 'Algorithmic content moderation: technical and political challenges in the automation of platform governance'. *Big Data & Society* 7(1). Available at: <https://journals.sagepub.com/doi/full/10.1177/2053951719897945>

Grimmelman, J. (2015). 'The Virtues of Moderation'. *Yale Journal of Law & Technology*. 7 (42). Available at: https://papers.ssrn.com/sol3/Papers.cfm?abstract_id=2588493

Hannák et al. (2017). 'Bias in online freelance marketplaces: evidence from TaskRabbit and Fiverr'. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. p1914–1933. Available at: <https://doi.org/10.1145/2998181.2998327>

Hill, K. (2021). 'The secretive company that might end privacy as we know it'. *The New York Times*. Available at: <https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html>

Koshiyama, A. et al. (2021) 'Towards algorithm auditing: a survey on managing legal, ethical and technological risks of AI, ML and associated algorithms.' Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3778998

Lee et al. (2015.) 'Working with Machines: The Impact of Algorithmic and Data-Driven Management on Human Workers'. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 1603-1612. Available at: <https://dl.acm.org/doi/abs/10.1145/2702123.2702548>

Leskovec, J., Rajaraman, A. and Ullman, J. D. (2014). 'Chapter 9: Recommendation Systems'. *Mining of Massive Datasets*. Cambridge University Press. Second edition. Available at: <https://www.cambridge.org/core/books/abs/mining-of-massive-datasets/recommendation-systems/8E2DDDAEFC644266620945386AB7DFDE>

Lum, K. and Chowdhury, R. (2021). 'What is an "algorithm"? It depends whom you ask.' *MIT Technology Review*. Available at: <https://www.technologyreview.com/2021/02/26/1020007/what-is-an-algorithm/>

Matthews, J. et al. (2019). 'The right to confront your accusers: Opening the black box of forensic DNA software'. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. p321–327. Available at: <https://doi.org/10.1145/3306618.3314279>

Mattu et al. (2021). 'How we built a Facebook inspector'. *The Markup*. Available at: <https://themarkup.org/citizen-browser/2021/01/05/how-we-built-a-facebook-inspector>

Microsoft. 'PhotoDNA'. *Microsoft.com*. Available at: <https://www.microsoft.com/en-us/photodna> [Accessed 10 November 2021].

Mikians, J. et al. (2013). 'Crowd-assisted search for price discrimination in e-commerce: first results'. *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, p1-6. <https://doi.org/10.1145/2535372.2535415>

Moss et al. (2021). 'Assembling accountability: algorithmic impact assessment for the public interest'. *Data & Society*. Available at: <https://datasociety.net/library/assembling-accountability-algorithmic-impact-assessment-for-the-public-interest/>

Narayanan, A. (2019). 29 December. Available at: https://twitter.com/random_walker/status/1211262520247439361

Noble, S. (2013). 'Google Search: Hyper-Visibility as a Means of Rendering Black Women and Girls Invisible'. In: *Visible Culture: An Electronic Journal for Visual Culture*. Available at: <http://ivc.lib.rochester.edu/google-search-hyper-visibility-as-a-means-of-rendering-black-women-and-girls-invisible/>

Ofcom. (2020) 'Covid-19 news and information: consumption, attitudes and behaviour'. *Ofcom*. Available at: <https://www.ofcom.org.uk/research-and-data/tv-radio-and-on-demand/news-media/coronavirus-news-consumption-attitudes-behaviour>

Pasquale, F. (2010). 'Beyond innovation and competition: the need for qualified transparency in internet intermediaries'. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1686043

Raji, D. and Buolamwini, J. (2019). 'Actionable auditing: investigating the impact of publicly naming biased performance results of commercial AI products'. *Association for the Advancement of Artificial Intelligence*. Available at: https://www.thetalkingmachines.com/sites/default/files/2019-02/aies-19_paper_223.pdf

Roose, K. (2019). 'The making of a YouTube radical'. *New York Times*. Available at: <https://www.nytimes.com/interactive/2019/06/08/technology/youtube-radical.html>

Sandvig, C. et al. (2014). 'Auditing algorithms: research methods for detecting discrimination on internet platforms'. In *Proceedings of 'Data and Discrimination: Converting Critical Concerns into Productive Inquiry', a preconference at the 64th Annual Meeting of the International Communication Association*, p1–23. Available at: <http://social.cs.uiuc.edu/papers/pdfs/ICA2014-Sandvig.pdf>

StrategyR. (2021). *Global Artificial Intelligence (AI) Governance Market to Reach \$402 Million by 2026*. Available at: <https://www.strategyr.com/market-report-artificial-intelligence-ai-governance-forecasts-global-industry-analysts-inc.asp>

Twitter. (n.d). *Twitter API documentation*. Available at: <https://developer.twitter.com/en/docs/twitter-api> [Accessed 11.11.2021].

UK Competition and Markets Authority. (2021). 'Algorithms: how they can reduce competition and harm consumers'. *Gov.uk*. Available at: <https://www.gov.uk/government/consultations/algorithms-competition-and-consumer-harm-call-for-information/algorithms-how-they-can-reduce-competition-and-harm-consumers>

United States District Court for the District of Columbia. (2020). 'Sandvig v. Barr – memorandum opinion'. *ACLU*. [online] Available at: <https://www.aclu.org/sandvig-v-barr-memorandum-opinion>

Wall Street Journal. (2021). *The Facebook Files: A Wall Street Journal Investigation*. Available at: <https://www.wsj.com/articles/the-facebook-files-11631713039>

Weber, M. S. and Kosterich, A. (2017). 'Coding the news: the role of computer code in filtering and distributing news'. *Digital Journalism*, 6:3, 310–329. Available at: <https://www.tandfonline.com/doi/abs/10.1080/21670811.2017.1366865?journalCode=rdij20>

Williams, J. (2018). 'D.C. court: accessing public information is not a computer crime'. *Electronic Frontier Foundation*. [online] Available at: <https://www.eff.org/deeplinks/2018/04/dc-court-accessing-public-information-not-computer-crime> [All accessed 11.11.21].

About the Ada Lovelace Institute

The Ada Lovelace Institute (Ada) is an independent research institute with a mission to make data and AI work for people and society.

We are working to create a shared vision of a world where AI and data are mobilised for good, to ensure that technology improves people's lives. We take a sociotechnical, evidence-based approach and use deliberative methods to convene and centre diverse voices. We do this to identify the ways that data and AI reorder power in society, and to highlight tensions between emerging technologies and societal benefit.

Ada was established by the Nuffield Foundation in early 2018, in collaboration with the Alan Turing Institute, the Royal Society, the British Academy, the Royal Statistical Society, the Wellcome Trust, Luminato, techUK and the Nuffield Council on Bioethics.

We are funded by the Nuffield Foundation, an independent charitable trust with a mission to advance social well-being. The Foundation funds research that informs social policy, primarily in education, welfare and justice. It also provides opportunities for young people to develop skills and confidence in STEM and research. In addition to the Ada Lovelace Institute, the Foundation is also the founder and co-funder of the Nuffield Council on Bioethics and the Nuffield Family Justice Observatory.

Find out more:

Website: [Adalovelaceinstitute.org](https://adalovelaceinstitute.org)

Twitter: [@AdaLovelaceInst](https://twitter.com/AdaLovelaceInst)

Email: hello@adalovelaceinstitute.org



Permission to share: This document is published under a creative commons licence: CC-BY-4.0

Preferred citation: Ada Lovelace Institute (2021).

Technical methods for the regulatory inspection of algorithmic systems in social media platforms. Available at: <https://www.adalovelaceinstitute.org/report/technical-methods-regulatory-inspection>

ISBN: 978-1-8382567-8-4