Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database

Sean McGregor
XPRIZE Foundation, Partnership on AI,* Syntiant Corp.
7555 Irvine Center Dr., Suite 200
Irvine, California 92618
iaai21@seanbmcgregor.com

Abstract

Mature industrial sectors (e.g., aviation) collect their real world failures in incident databases to inform safety improvements. Intelligent systems currently cause real world harms without a collective memory of their failings. As a result, companies repeatedly make the same mistakes in the design, development, and deployment of intelligent systems. A collection of intelligent system failures experienced in the real world (i.e., incidents) is needed to ensure intelligent systems benefit people and society. The AI Incident Database is an incident collection initiated by an industrial/non-profit cooperative to enable AI incident avoidance and mitigation. The database supports a variety of research and development use cases with faceted and full text search on more than 1,000 incident reports archived to date.

Introduction

Governments, corporations, and individuals are increasingly deploying intelligent systems to safety-critical problem areas, including transportation (NTSB 2017) and law enforcement (Dressel and Farid 2018), as well as challenging social system domains such as recruiting (Dastin 2018). Failures of these systems pose serious risks to life and wellbeing, but even good-intentioned intelligent system developers fail to imagine what can go wrong when their systems are deployed in the real world. Worse, the artificial intelligence system community has no formal systems whereby practitioners can discover and learn from the mistakes of the past. Individuals in technology (Olsson 2019; Lutz 2020), legal practice (Hall 2020), and reputation management (Pownall 2020) now collect artificial intelligence failure history on Google Docs and GitHub. While these are admirable efforts, a person checking for problems matching their technology or problem domain will need to page through lists of links to find ones of potential relevance. Existing lists are difficult to use in development, are not comprehensive archives, and are representative of individual viewpoints of artificial intelligence (AI) failures in the real world.

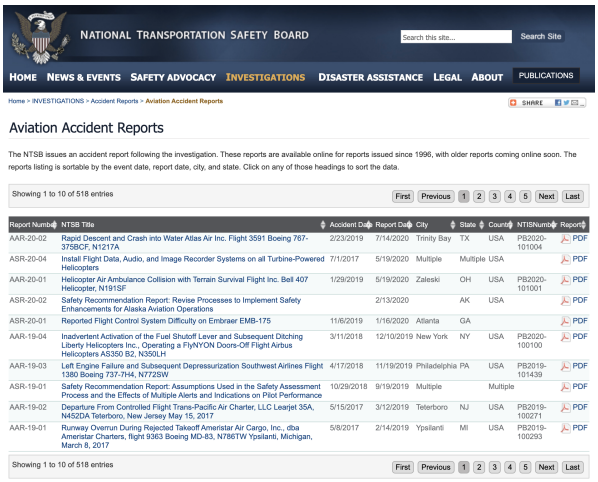

Figure 1: The US National Transportation Safety Board's (NTSB's) database shown above indexes incident and accident timelines, location, meteorology, severity, aircraft, operators, and phase of flight as facets. The reports also have a full text narrative that is searchable. Upon completion of an investigation, the report is indexed alongside the case record within the database (Federal Aviation Administration 2020).

Avoiding repeated AI failures requires making past failures known to AI practitioners. Therefore, we introduce a systematized collection of incidents where intelligent systems have caused safety, fairness, or other real world problems. The AI Incident Database (AIID) answers the question, "what can go wrong when someone deploys this system"?

The contributions of this work are three fold. We provide infrastructure supporting best practices within the artificial intelligence industry, a dataset of more than one thousand incident reports, and an architecture for building research products on the growing collection of incidents. We begin by exploring incident databases in other fields of practice before introducing the system architecture of the AIID. We then wrap up with a few concluding remarks.

*Representing the XPRIZE Foundation as a Partnership on AI non-profit partner.

Other Incident Databases

Many industries have their own industry-specific incident databases, including computer security (The MITRE Corporation 2020), aviation (Federal Aviation Administration 2020; National Aeronautics and Space Administration 2020), and medicine (United States Food and Drug Administration 2020). Each play important roles in avoiding or mitigating potential harms in their respective industries, but in particular the aviation and computer security examples inspire the AIID.

The commercial air travel industry owes much of its increasing safety to systematically analyzing and archiving past accidents and incidents within a shared database. In aviation, an accident is a case where substantial damage or loss of life occurs. Incidents are cases where the risk of an accident substantially increases. For example, when a small fire is quickly extinguished in a cockpit it is an "incident" but if the fire burns crew members in the course of being extinguished it is an "accident." The aviation database (see Figure 1) indexes flight log data and subsequent expert investigations into comprehensive examinations of both technological and human factors. In part due to this continual self-examination, air travel is one of the safest forms of travel. Decades of iterative improvements to safety systems and training have decreased fatalities 81 fold since 1970 when normalized for passenger miles (Mediavilla 2020).

Aviation accidents share a well-defined operational context, but intelligent systems can be applied to all contexts. The comprehensive nature of "intelligence" means AI incident databases ingest unforeseen and novel contexts, technologies, and failures. The AIID design outlined in the next section introduces a system architecture inspired by the aviation incident and accident database but with a greater emphasis on extensibility.

The second incident database inspiring the AIID is the Common Vulnerabilities and Exposures (CVE) system, which contains $141,076$ publicly disclosed cybersecurity vulnerabilities and exposures (The MITRE Corporation 2020). In contrast to the aviation database, which serves users associated with a single industry, the CVE site serves as critical security infrastructure across all industries by enabling vulnerabilities to be circulated and referenced with a consistent identifier. Other systems build on the identifiers with taxonomies (e.g., the Common Vulnerability Scoring System), produce research, and develop more secure software. The creation of numbered identifications forms community infrastructure that the field of artificial intelligence currently lacks. The lists of Olsson (2019), Lutz (2020), Hall (2020), and Pownall (2020) lack the comprehensive coverage, identification, and extensibility properties of the CVE, and the full text search capability of the NTSB database.

The AI Incident Database

The AIID defines an "AI incident" as a situation in which AI systems caused, or very nearly caused, real-world harm. A more extensive exploration of AI incident definition is provided in the AIID's documentation (McGregor and Arnold 2020). Applying the definition led to the indexing of more than 1,000 publicly available "incident reports," which are a mixture of documents from the popular, trade, and academic press. Multiple reports often pertain to a single incident collectively joined together by a single identifier. For example, incident number 3 is composed of 18 reports on the Boeing 737 MAX 8 crashes (Olsson 2018a). The variety of reports serves several purposes. First, it provides multiple viewpoints on incidents for which there is often disagreement about fair characterizations. In the Boeing case, people disagree on the extent to which technological or human factors played a part in the tragedies. Second, the number of publications and publication types serves as a proxy for interest in the incident. More reported incidents are typically more damaging, more sensational, or both. After opening the AIID to public submissions, we expect incident 3 will have thousands of incident reports due to intense public interest in the safety of flight. Lastly, sampling multiple reports per incident gives more complete coverage of words associated with an incident and increases the likelihood of users discovering incidents relevant to their use cases. The use cases are detailed in the following user stories.

User: Product Managers. Corporate product managers are responsible for defining product requirements before and during product development. If a product manager discovers incidents where intelligent systems have caused harms in the past, they can introduce product requirements to mitigate risk of recurrence. For example, when a product manager is specifying a recommender system for children, the AIID should facilitate the discovery of incident 1 (Yampolskiy 2020), wherein YouTube Kids recommended inappropriate content. Knowledge of incident 1 would produce a range of technological, marketing, and content moderation requirements for the product.

User: Risk Officers. Organizationally, risk officers are tasked with reducing the strategic, reputational, operational, financial, and compliance risks associated with an enterprise's operation. Consider the case of a social network preparing to launch a new automatic translation feature. A search of "translate" within the AIID returns 40 separate reports, included among them an incident wherein a social media status update of "good morning" translated to "attack them" and resulted in the user's arrest (Anonymous 2017). After discovering the incident, the risk officer can read reports and analyses to learn that it is currently impossible to technologically prevent this sort of mistake from happening, but there is a variety of best practices in mitigating risk, such as clearly indicating the text is a machine translation.

User: Engineers. Engineers can also benefit from checking the AIID to learn more about the real world in which their systems are deployed. Consider the case of an engineer who is making a self-driving car with an image recognition system. The experience of incident 36 (Olsson 2018b), where a woman in China was shamed for jaywalking because her picture was on the side of a bus, shows how images can confuse image recognition systems. Such cases must therefore be represented within safety tests.

User: Researchers. Safety and fairness researchers already employ case study methodologies in their scholarship (Yampolskiy 2019; Scott and Yampolskiy 2019), but

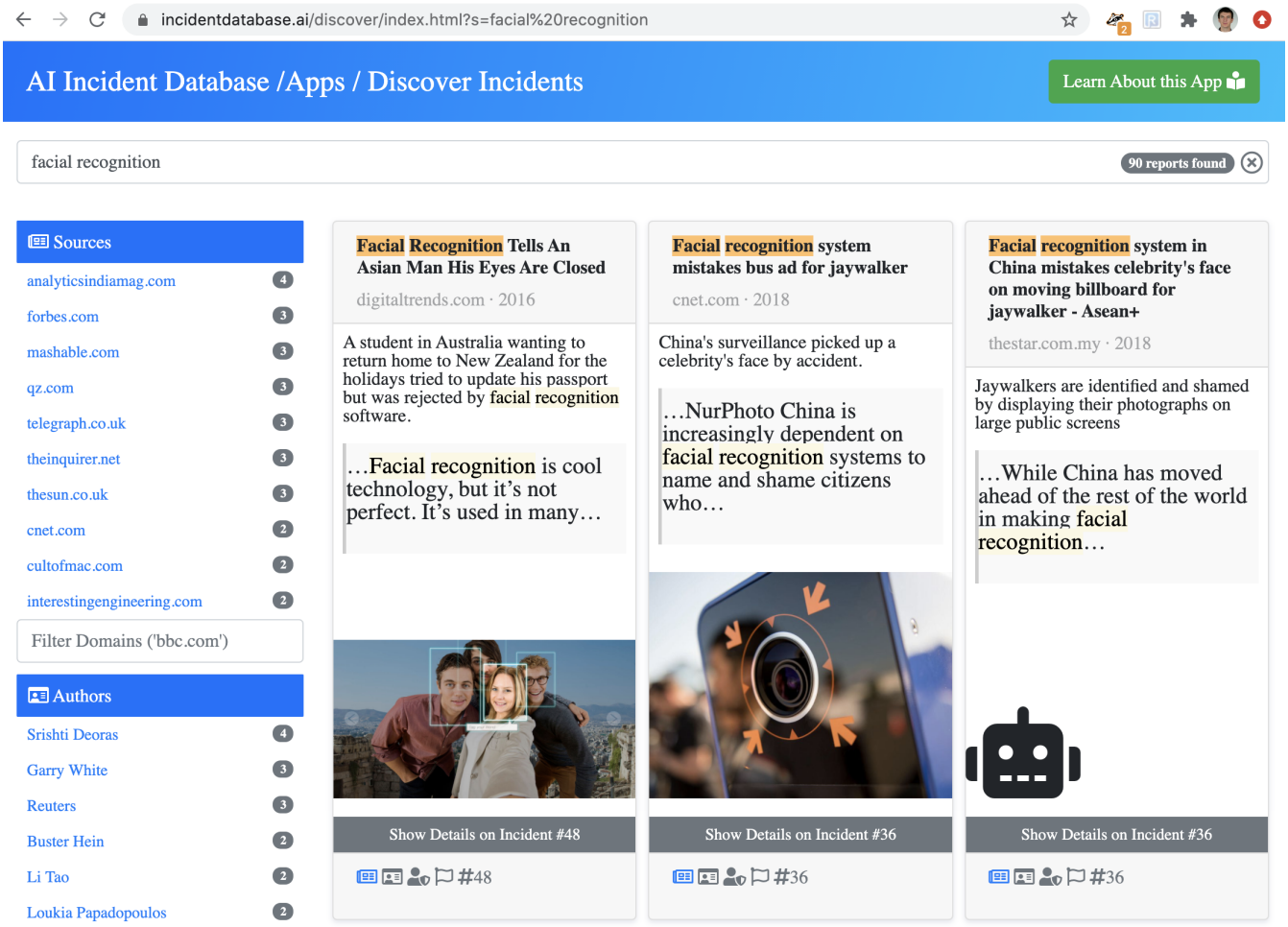

Figure 2: A user has entered "facial recognition" as a search term into the search box of the "Discover" AIID application. 90 reports returned to the search instantaneously (every keystroke filters the results and the page renders) and the matching text from the reports is snippeted. The publications represented within the results are faceted in the left column along with the authors, submitters, and incident numbers to support filtering the reports based on their metadata.

they presently lack the capacity to track AI incidents at the population level. For example, it is difficult to show the rate at which incidents involving policing are changing through time. An AIID search for "policing" in the full text of reports currently returns 14 distinct incidents. Each of these incidents are additionally citeable within research papers. The resulting research papers can then be added to the database as further reporting on the incident. Additionally, researchers can show the importance of their publications by citing incidents that could potentially be mitigated through their advances.

Finally, we note that making a database entry shareable (i.e., linkable) empowers these users rhetorically to convince others that mitigation is necessary. Technology companies are famous for their penchant to move quickly without evaluating all potential bad outcomes. When bad outcomes are enumerated and shared, it becomes impossible to proceed in ignorance of harms.

System Architecture

The AIID is sponsored by the Partnership on AI (PAI), which is a multi-stakeholder organization funded by technology companies and governed by a board of directors split between corporate partners and non-profit civil society organizations. Much of the system architecture is motivated by serving the varied interests and viewpoints of PAI members, which often are in diametric opposition to one another. While convergence of views is not expected, exposure to a diverse set of views may lead to a more holistic understanding of incident impacts among the AIID's users. Ingesting multiple reports per incident provides diverse viewpoints on incidents, but so too should the system architecture be amenable to multiple viewpoints of reports. At its core, the database is a MongoDB document database storing incident report text and metadata, but the associated build pipeline supports multiple statically hosted applications and data summaries that integrate with one another via taxonomies.

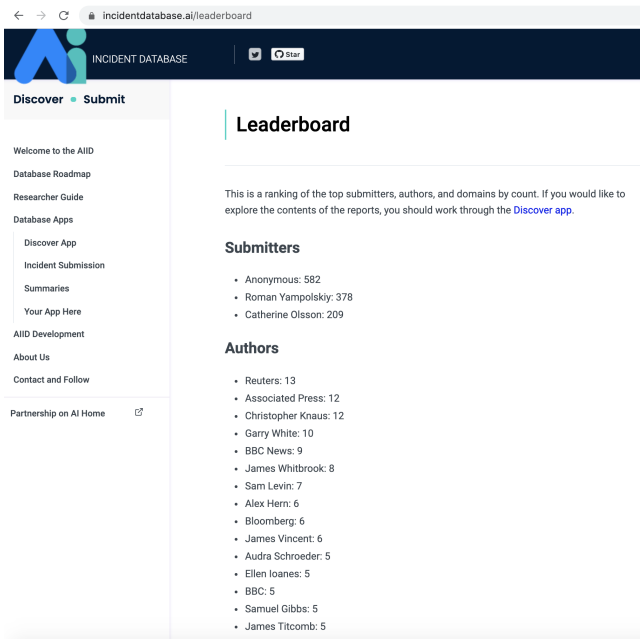

Figure 3: The database provides a leaderboard of submitters and authors totalling the number of reports associated with their submissions. Gamification in other contexts has shown that people are more eager to volunteer their time for a community resource if the sum total of their contribution is constantly recognized and reinforced.

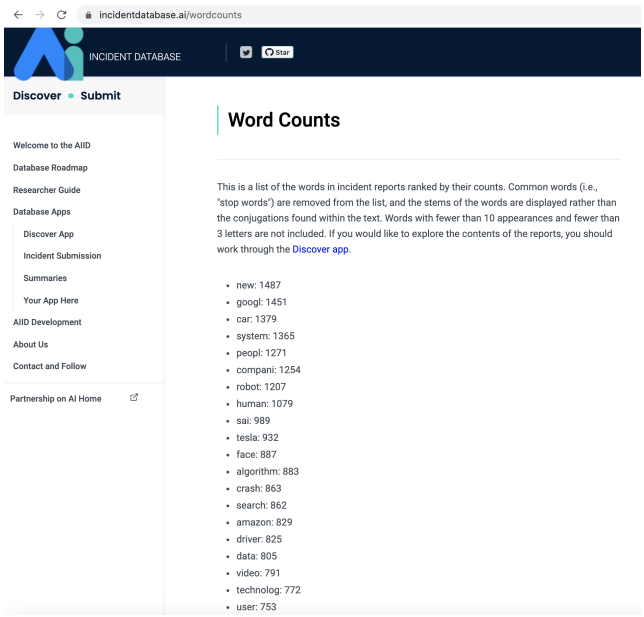

Figure 4: Every time the applications rebuild, the complete text of all reports are queried, stemmed, and stop words are removed. The top words in the database are then rank ordered and rendered in the page. This application generates no requests to the database when a user requests it since the page is pre-built in the application rendering pipeline. This means computationally expensive natural language models could be applied in the application lifecycle (e.g., training topic models) without negatively impacting user experience.

AIID Applications are responsible for actively managing and querying data within the database. The first application developed for the database is the "Discover" application, which is built to help users discover past incidents relevant to their work. Figure 2 shows one search in the Discover application. All searches in the Discover application issue Algolia "instant searches," meaning they return results in less than a second. By offloading the instant search functionality to a secondary index, the Discover application's heavy database queries cannot negatively impact other applications in the AIID. Another application is the "Submit" application, which is a form for submitting links to publicly available incident reports. The Submit application supports the incident ingestion process by checking the reports against reports already found in the database.

Where applications actively query and modify the database, "data summaries" are static snapshots of the database at the time they are generated (see Figures 3 and 4). The problem with these database views is that they often require iterating over the complete database. If these pages render for the user every time the user visits the page, the database would be slow and expensive to host. Instead, the AIID periodically pre-renders database views as static web applications, which means they only require a single database request at the time the website builds. As such, it is possible to develop a gallery of views into the data similar to the D3JS gallery, which has 168 different visualization examples (Bostock 2020). Similar visualizations are planned for the AIID for trend analysis, including with topic models and structured reports that monitor technology, affected populations, or problem domains through time. These analyses can be incorporated into the static build (see Figure 5) and update automatically when the website updates.

All incident reports have metadata captured on entry into the database, including title, source, author, submitter, publication date, incident date, and incident number. These are all objective facts that can be filtered as shown in Figure 2. Where applications such as the Discover application filters these objective facts, it also has the capacity to filter based on subjective taxonomic classification of reports and incidents. Taxonomies are granted namespaces managed by individuals or organizations, who are not required to maintain a global consensus. This avoids the challenge of developing a single shared universal ontology for AI incidents and instead allows for multiple viewpoints on the data to develop and compete for mindshare. Database applications manage their own taxonomies, but all applications and data summaries may consume taxonomies for their functionality and reporting (see Figure 5). While the classifications are all controlled by their own application and managing entity, they can be applied as filters across all applications within the AIID.

When developers push code to the AIID GitHub repository, applications that hook into the database are statically

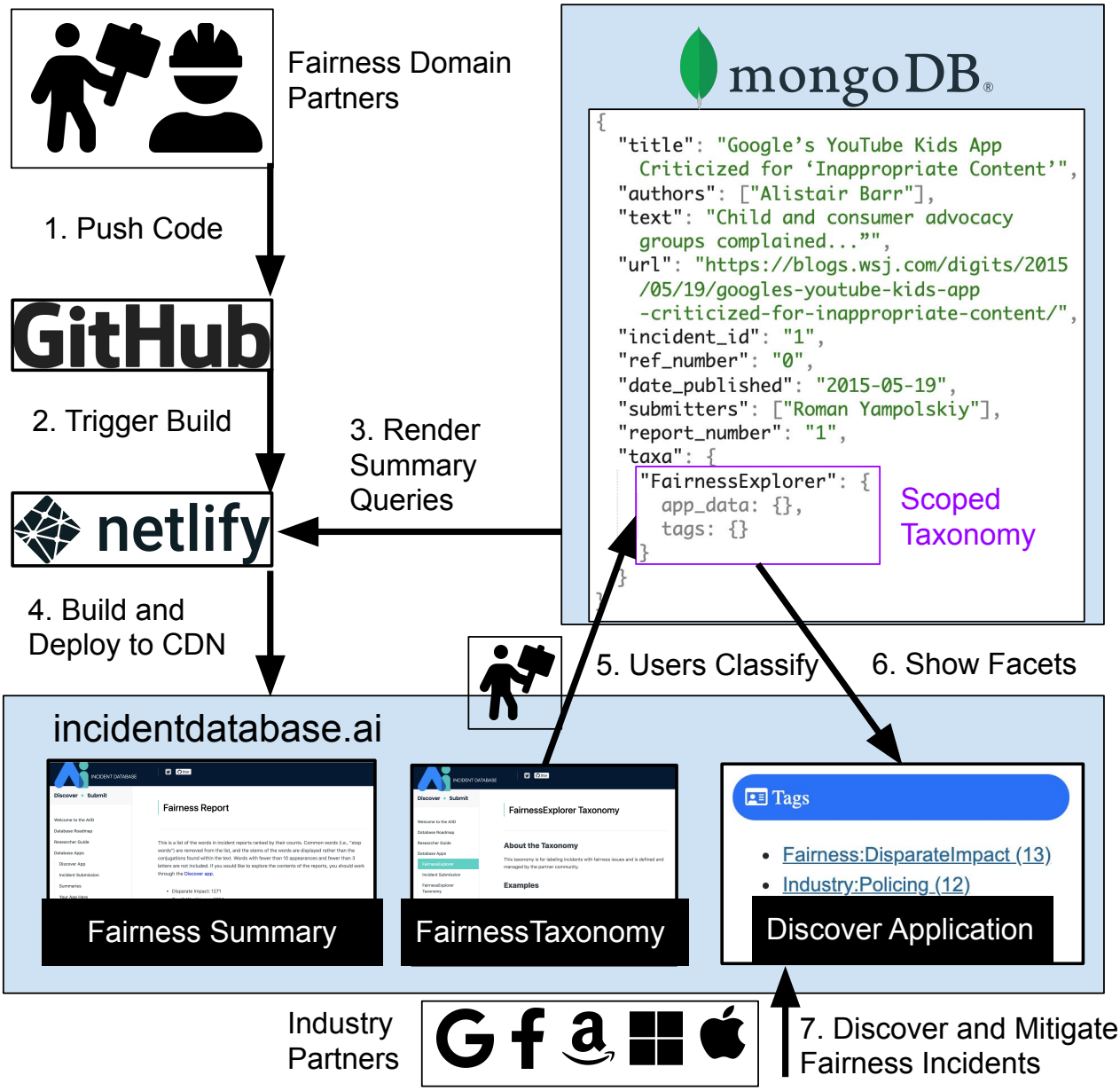

Figure 5: Presuming a civil society organization would like to highlight the fairness properties of incidents within the database, they are able to extend the database with a fairness taxonomy that is consumable by all applications and data summaries of the AIID. First, a definition of the taxonomy and a user interface for managing it are developed (see the "Fairness Taxonomy" above containing both). The goal of the Fairness Taxonomy application is to produce inter rater reliability for incident reports by supporting classifications with documentation and programmatic tools. The civil society organization also defines a Fairness Summary, which will programmatically generate through periodic refreshes on the fairness taxonomy data. Whenever the civil society organization wishes to update their application and summaries, they push code to GitHub and trigger a build on the static website hosting service Netlify. The build process queries the database to generate static summaries of the database contents, including the Fairness Summary. The website then deploys to a global content distribution network. Users can then apply classifications within the namespace of the Fairness Taxonomy. When industry partners visit the website, they can filter incidents in the Discover application based on the classifications of the Fairness Taxonomy.

rendered and deployed by the AIID hosting provider. Since the web server does not render the web application at request time, it can service very large user volumes. Further, the absence of dynamic code in the server means multiple versions of the AIID front end can be hosted simultaneously at negligible cost.

Conclusion

We expect the extensible architecture will provide for the most pragmatic coverage of AI incidents through time while reducing negative consequences from AI in the real world. Early indications of adoption are strong. Even prior to publishing the database, we received collaboration requests from "Big 4" accounting firms, international consultancies, law firms, research institutes, and individual academics. Through time we hope the database will develop from the work product of a small team of individuals into community owned infrastructure aligned with producing the most beneficial intelligent systems for people and society. To quote Santayana, "Progress, far from consisting in change, depends on retentiveness... Those who cannot remember the past are condemned to repeat it." (Santayana and Cory 1924)

Acknowledgments

The author is grateful to Jingying Yang and the ABOUT ML project at the Partnership on AI for sponsoring this work and to the incident submitters, especially to Roman Yampolskiy, Catherine Olsson, Sam Yoon, Patrick Hall, Charlie Pownall, Zachary Arnold, Ingrid Dickinson, Thomas Giallella, Nicolina Demakos, Lawrence Lee, and Darlena Phuong Quyen Nguyen for their efforts in collecting many incidents to date. Iftekhar Ahmed and Ben Shneiderman also provided helpful feedback on the paper.

References

Anonymous. 2017. Incident Number 72. *AI Incident Database* URL https://incidentdatabase.ai/cite/72. Retrieved December 28, 2020.

Bostock, M. 2020. Gallery. URL https://observablehq.com/@d3/gallery. Retrieved December 28, 2020.

Dastin, J. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. URL https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G. Retrieved December 28, 2020.

Dressel, J.; and Farid, H. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science Advances* 4(1): 1–6. ISSN 23752548. doi:10.1126/sciadv.aao5580.

Federal Aviation Administration. 2020. Accident and Incident Data. URL https://www.faa.gov/data_research/accident_incident/. Retrieved December 28, 2020.

Hall, P. 2020. awesome-machine-learning-interpretability. *GitHub Pages* URL https://github.com/jphall663/awesome-machine-learning-interpretability/blob/master/README.md#ai-incident-tracker. Retrieved December 28, 2020.

Lutz, R. 2020. Learning from the past to create Responsible AI. *GitHub Pages* URL https://romanlutz.github.io/ResponsibleAI/. Retrieved December 28, 2020.

McGregor, S.; and Arnold, Z. 2020. Researcher Guide. *AI Incident Database* URL https://incidentdatabase.ai/research. Retrieved December 28, 2020.

Mediavilla, J. I. 2020. Aviation safety evolution (2019 update). URL https://theblogbyjavier.com/2020/01/02/aviation-safety-evolution-2019-update/. Retrieved December 28, 2020.

National Aeronautics and Space Administration. 2020. Aviation Safety Reporting System. URL https://asrs.arc.nasa.gov/.

NTSB. 2017. Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor-Semitrailer Truck. *Highway Accident Report* 63. ISSN 1473-6691. doi:10.1093/jicru/ndl025. URL https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR1702.pdf.

Olsson, C. 2018a. Incident Number 3. *AI Incident Database* URL https://incidentdatabase.ai/cite/3. Retrieved December 28, 2020.

Olsson, C. 2018b. Incident Number 36. *AI Incident Database* URL https://incidentdatabase.ai/cite/36. Retrieved December 28, 2020.

Olsson, C. 2019. Tweet About List Keeping. URL https://twitter.com/catherineols/status/1105561165646585857. Retrieved December 28, 2020.

Pownall, C. 2020. AI and algorithmic incidents and controversies. URL https://charliepownall.com/ai-algorithimic-incident-controversy-database/. Retrieved December 28, 2020.

Santayana, G.; and Cory, D. 1924. *The Life of Reason: Or, The Phases of Human Progress*. C. Scribner's sons.

Scott, P.; and Yampolskiy, R. 2019. Classification Schemas for Artificial Intelligence Failures. *Delphi - Interdisciplinary Review of Emerging Technologies* 2(4): 186–199. ISSN 26263734. doi:10.21552/delphi/2019/4/8. URL http://delphi.lexxion.eu/article/DELPHI/2019/4/8.

The MITRE Corporation. 2020. CVE - Common Vulnerabilities and Exposures. URL https://cve.mitre.org/. Retrieved December 28, 2020.

United States Food and Drug Administration. 2020. FDA Adverse Event Reporting System (FAERS) Public Dashboard. *Food and Drug Administration* URL https://www.fda.gov/drugs/guidancecomplianceregulatoryinformation/surveillance/adversedrugeffects/ucm070093.htm. Retrieved December 28, 2020.

Yampolskiy, R. V. 2019. Predicting future AI failures from historic examples. *Foresight* 21(1): 138–152. ISSN 14636689. doi:10.1108/FS-04-2018-0034.

Yampolskiy, R. V. 2020. Incident 1. *AI Incident Database* URL https://incidentdatabase.ai/cite/1. Retrieved December 28, 2020.