FAT-Forensics: A Python Toolbox for Fairness, Accountability and Transparency Forensics

Anonymous Author(s)
Affiliation
Address
email

Abstract

Reproducibility in artificial intelligence and machine learning research is a well known issue. One approach to battle it is open source software that allows other researchers to compare and contrast their novel approach against other implementations without the burden of re-implementing them. In this paper we discuss a fragmented landscape of open source tools in the Fairness, Accountability and Transparency (FAT) research field and propose an open-sourced Python framework, called FAT-Forensics, that aims at providing a common research platform in this area.

1 Introduction

Open source software is a backbone of reproducible research. Especially when considering the nature of artificial intelligence (AI) and machine learning (ML) algorithms where sometimes changing the seed of a random number generator can cause a state-of-the-art solution to become a sub-par one. Both these fields are struggling with reproducibility crisis [1]. Whether it is due to poor reporting, trade secrets or simply trying to keep an advantage over a competition remains an open question. One way to tackle this problem is to promote publishing the software used for scientific experiments under an open source licence or simply require it as a part of the publishing and peer-review process, which has been advocated for a long time [2]. Despite their importance, implementations are commonly treated just as a research by-product and are often abandoned after publishing a work based on them. Alternatively, they are provided as a standalone package that often does not follow the best software engineering practices, hence can prove difficult to use for a wider community due to lack of documentation, updates and maintenance, therefore impacting its usability and reproducibility in general.

To help mitigate such undesired practices in the field of artificial intelligence and machine learning Fairness, Accountability and Transparency (FAT) we propose a FAT-Forensics open source Python package in section [2]. It is intended to be a framework for implementing and testing novel algorithms invented by the FAT research community and facilitate their evaluation and comparison against state-of the art ones. We also discuss a variety of relevant and freely available software packages that can be used to assess security, privacy, fairness, interpretability and explainability of data processing pipelines, namely (raw) data, features, predictive models and algorithmic decisions in section [3]. Finally, we conclude this position paper with a discussion and envisaged development timeline in section [4].

2 FAT-Forensics

Depending on the maturity of a research field proposing, adopting and developing a common software infrastructure for implementing novel algorithms and comparing them with others may be difficult.
Relatively young and still shaping fields, like FAT of predictive systems, usually lack this type of software solutions. Within the past few year we have seen increasing number of novel FAT algorithms implemented in different programming languages, each one with a different requirements and Application Programming Interface (API) making them difficult to compare in a systematic way. To address this issue, while the community is still young and flexible enough to adopt it, we propose design and implementation plan of an open source Python framework for evaluating and comparing FAT algorithms. We decided to develop it in Python given its popularity in AI and ML research communities and its overall simplicity. We call this toolbox FAT-Forensics and we will release it in early 2019 under 3-clause BSD License, therefore opening it up for commercial applications. It will collate all of the state-of-the-art FAT algorithms implementation and provide a coherent API to make it accessible to the community. For relevant software packages that are already well established in the FAT community we will provide wrappers, hence avoid re-implementing them and make them accessible under a common API. Furthermore, we will encourage researchers and practitioners alike to contribute their novel approaches to our software package or make them compatible with it. We will enable the community to contribute their solution effortlessly, i.e. without extensive software engineering training, by putting a dedicated process in place. We hope that all these steps will help us create a sustainable software package that is easy to extend and will serve the community for a long time. Last but not least, we hope that the package will help to mitigate all the potential reproducibility issues that otherwise could have arisen in the FAT community in the future as our package aims to become a test-bed for implementing, testing, evaluating and sharing cutting-edge FAT research.

The development of the package is currently focused on three main use-cases. First of all, we want to integrate existing interpretability packages – discussed in section 3.2 – under a common API. We believe that re-implementing high quality interpretability and explainability techniques that are already made available to the community is unnecessary, however to make our package more versatile and appealing we will provide access to them (a wrapper) via a unified API. Secondly, we will implement the most common fairness metrics – e.g. group unaware, equal opportunity, equal accuracy, demographic parity [3] – and showcase their use on the UCI Census Income data set (the data set of choice in algorithmic fairness research) and to recreate the arguments fuelling the discourse between ProPublica and Northpointe about the fairness and racial bias of the COMPAS system[1]. Thirdly, we will implement basic data set anonymisation algorithms, like k-anonymity, l-diversity and t-closeness [4], given lack of their open source implementations to help our users assess privacy and security of their data sets. Finally, we will open up the package to community contributions as this will make it possible for all the researchers to extend it with cutting-edge FAT algorithms and expose them to the wider community within a hand’s reach.

3 Related Work

In the United Kingdom a special academic body – called the Software Sustainability Institute[7] – has been created with the sole purpose of educating the research community about the best software engineering practices. The main reason behind its inception is a general belief that better software means better (and more reproducible) research, which has proven to be true in many areas of AI and ML research. Nevertheless, sometimes it is difficult to discourage dubious software engineering practice especially when a programming language of choice – Python being the most popular one in the AI and ML communities given its easy syntax, wealth of packages and prototyping capabilities – does not discourage it considering its properties like being dynamically typed.

3.1 AI Community Effort

In well established research communities, e.g. supervised learning or reinforcement learning, a consensus among researchers is arising and each community is converging towards using a common performance metric or an evaluation software framework. For predictive performance of supervised learning algorithms these can be, for example: accuracy, F1 score or AUC, which are a compulsory component of any such software framework – cf. sklearn.metrics sub-module in the scikit-learn package [5] and tf.metrics in the TensorFlow package [6]. Given the independence of these

http://archive.ics.uci.edu/ml/datasets/Census+Income
https://www.technologyreview.com/s/607955/inspecting-algorithms-for-bias/
https://www.software.ac.uk/
metrics from the underlying ML algorithm implementation some software, e.g. PyCM[^1], is focused
entirely on calculating these metrics. In other research environments, e.g. reinforcement learning,
there are common software platforms that are used to systematically compare (train and evaluate)
novel approaches, hence making research results easy to reproduce and compare. Examples of these
are: Project Malmo[^8], OpenAI Gym[^9] and OpenAI MuJoCo[^10]. Alternatively, projects such
as cookiecutter[^5] allow researchers to create software packages that follow common structure, hence
make them easier to execute and integrate.

3.2 Fairness, Accountability and Transparency

FAT research software landscape is more scattered in comparison to maturing fields such as supervised
learning. A recent attempt to create a common framework for FAT algorithms is the What-If
tool[^6] – with fairness model evaluation and counterfactual prediction explainability functionality
– released by Google as a plugin to their interactive ML model analytic tool called TensorBoard.
Despite its similarity to our project the tool is only compatible with TensorFlow models which is
a serious limitation. In addition to a general framework such as the What-If tool we can also find
implementations of particular interpretability and explainability algorithms published in the literature.
Examples of these are: LIME[^7], Anchor[^8] and PyCEbox[^9]. Many of these have been
collected and built into packages with the most prominent ones being: Skater[^10], eli5[^11] and
shap[^12]. As discussed earlier we do not want to re-implement these, therefore our package will contain
a wrapper for all of them with the aim of unifying their APIs. Open source software landscape of
fairness in AI and ML is even more irregular. The availability of fairness packages is limited, they
use different programming languages, often lack a licence or documentation and vary in code quality.
The most prominent ones are: BlackBoxAudining[^13], fairness-comparison[^14], fairness[^15],
fairml[^16] and fairlearn[^17]. Finally, open source software for ML and AI security is next to
non-existent with one-off implementations of selected algorithm published on the Internet.

4 Conclusions

Despite software being the primary driver of progress in AI and ML research, its quality is often
less than desirable. Some research fields such as supervised learning and reinforcement learning
have reached consensus on that matter and have standardised metrics or software frameworks
used to evaluate and compare novel algorithms. On the other hand, fairness, accountability and
transparency research in AI and ML community lacks a common software infrastructure to analyse
and communicate the results of our research in a coherent manner. In this paper we proposed a
flexible open source Python tool to facilitate easy evaluation and comparison of FAT algorithms. We
envision the platform to have easy to use contribution mechanism to encourage other researchers to
release their software as a module for the proposed here software package instead of a standalone
code, hence abate the community effort to replicate and compare state-of-the-art approaches.

References


[^1]: https://github.com/sepandhaghighi/pycm
[^2]: https://github.com/audreyr/cookiecutter
[^3]: https://pair-code.github.io/what-if-tool/
[^4]: https://github.com/marcotcr/lime
[^5]: https://github.com/marcotcr/anchor
[^6]: https://github.com/AustinRochford/PyCEbox
[^7]: https://github.com/datascienceinc/Skater
[^8]: https://github.com/TeamHG-Memex/eli5
[^9]: https://github.com/slundberg/shap
[^10]: https://github.com/algofairness/BlackBoxAuditing
[^11]: https://github.com/algofairness/fairness-comparison
[^12]: https://github.com/columbia/fairtest
[^13]: https://github.com/adebayoj/fairml
[^14]: https://github.com/Microsoft/fairlearn


