PrivFair: A Library for Privacy-Preserving Fairness Auditing

Sikha Pentyala[1], David Melanson[2], Martine De Cock[2], Golnoosh Farnadi[1]

{sikha.pentyala,gfarnadi}@mila.quebec, {melanson,mdecock}@uw.edu

[1]Mila - Quebec AI Institute [2]School of Engineering and Technology, University of Washington Tacoma

W UNIVERSITY of WASHINGTON

Mila

Motivation

Biased predictions by machine learning (ML) models can lead to:

- Disparity and discrimination
- Withholding opportunities and resources

Auditing a model can help mitigate such biases

Problem

- Models can be protected and private to model owners
- Sensitive auditing data cannot be revealed to third party due to privacy regulations

Solution

PrivFair

Significance

PrivFair

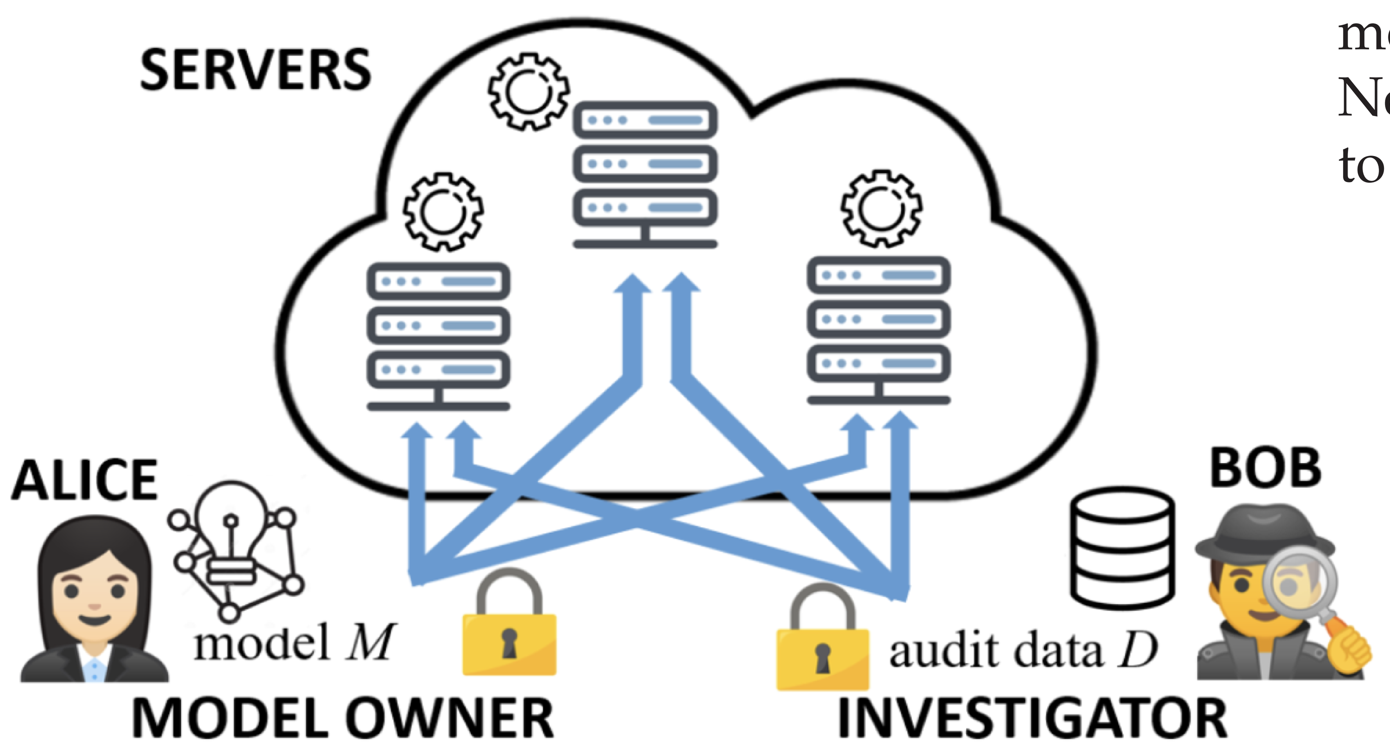
- Finds its use in AI services that deal with sensitive data such as in healthcare, banking, predictive policing etc. where one not only needs to protect data but also ensure that unbiased AI services are available to all.
- First step where AI developers and AI end-users can collaborate to aim for unbiased AI services, while protecting their data.
- Can be significant in all fields where ML models appear for automated decision making, e.g., education, housing, law-enforcement, healthcare, and banking, as well as new application domains yet to be discovered.

Conclusion

- **First-of-its-kind** library for privacy-preserving fairness audits of ML models.
- Longer runtimes are a price worth paying for our privacy, especially in fairness auditing where fast response times matter less.

Methodology

Computing over the data that one can not see. We use **Secure Multiparty Computation (MPC)** to enable servers to jointly compute a specified output (the fairness metric) from secret-shared information (**Alice**'s model and **Bob**'s audit data) in a distributed way, without learning the private information.



Secure fairness auditing in the 3PC scenario

Alice and **Bob** send shares of their data and model parameters to the servers.
Next, the servers run PrivFair's **MPC protocols** to

1. obtain shares of the label for each instance in Bob's audit data as classified with Alice's model
2. obtain shares of the elements of the confusion matrix for the above inferences: #true positives (TP), #false positives (FP), #true negatives (TN), #false negatives (FN)
3. obtain shares of statistical notions of fairness: demographic parity (**DP**), equalized odds (**EOD**), equal opportunity (**EOP**) and sub-group accuracy (**SACC**)

- **Demographic Parity.** A classifier satisfies **DP** if $P(\hat{Y}=1 \mid A=1) = P(\hat{Y}=1 \mid A=0)$.

$$\frac{\mathrm{TP}_{A=1} + \mathrm{FP}_{A=1}}{\mathrm{N}_{A=1}} = \frac{\mathrm{TP}_{A=0} + \mathrm{FP}_{A=0}}{\mathrm{N}_{A=0}}$$

- **Privacy.** No server learns secret-shared value x based only on the share of x that it receives. Throughout this process the servers compute on shares only, learning nothing about the values of the data, nor the model parameters, nor the fairness of the model.

- **Getting computed metrics.** In the final step, the servers send the computed shares of the fairness metrics to Bob who combines them to learn the result.

- **2PC and 3PC.** In addition to the 3PC scenario above, PrivFair[a] also supports the 2PC scenario in which Alice and Bob have enough resources to run the above protocols themselves.

[a]Implemented on top of MP-SPDZ (https://github.com/data61/MP-SPDZ); Available at https://bitbucket.org/uwtppml/privfair

Protocol π_{DP} for computing demographic parity (DP)

Input: The parties have a secret sharing of trained model parameters $[\![\mathcal{M}]\!]$, and a secret sharing of a data set $[\![\mathcal{D}]\!]$ with N instances and secret sharings $[\![Y]\!]$ and $[\![A]\!]$ of the corresponding ground truth labels and a binary sensitive attribute.
Output: A secret sharing of the DP metrics

```
[[Y_pred]] ← π_INFER([[M]],[[D]])
for i = 1 to N do
    [[N_A=1]] ← [[N_A=1]] + [[A[i]]]
end for
[[N_A=0]] ← N − [[N_A=1]]
for i = 1 to N do
    [[grnd]] ← π_EQ([[Y[i]]], 1)
    [[pred]] ← π_EQ([[Y_pred[i]]], 1)
    [[a]] ← [[A[i]]]
    [[tp]] ← π_MUL([[grnd]], [[pred]])
    [[ta]] ← π_MUL([[grnd]], [[a]])
    [[pa]] ← π_MUL([[pred]], [[a]])
    [[tpa]] ← π_MUL([[tp]], [[a]])
    [[TP_A=1]] ← [[TP_A=1]] + [[tpa]]
    [[FP_A=1]] ← [[FP_A=1]] + ([[pa]] − [[tpa]])
    [[TP_A=0]] ← [[TP_A=0]] + ([[tp]] − [[tpa]])
    [[FP_A=0]] ← [[FP_A=0]]
                 + ([[pred]] − [[pa]] − [[tp]] + [[tpa]])
end for
[[POS_A=0]] ← [[TP_A=0]] + [[FP_A=0]]
[[POS_A=1]] ← [[TP_A=1]] + [[FP_A=1]]
[[DP_A=0]] ← π_DIV([[POS_A=0]], [[N_A=0]])
[[DP_A=1]] ← π_DIV([[POS_A=1]], [[N_A=1]])
return [[DP_A|0,1]]
```

Runtime Results

Tabular Data. Binary classification –
Data source: German Credit Score dataset
Sensitive attribute: Gender
Model to audit: **log. regr. (47 params)**
Audit data: **200 samples**

		DP	EOP
Passive	2PC	3.34 sec	3.36 sec
	3PC	**1.37 sec**	1.67 sec
Active	2PC	239.24 sec	238.69 sec
	3PC	6.41 sec	6.43 sec

Images. Multi-class classification – 7 classes
Data source: RAVDESS emotions dataset
Sensitive attribute: Gender
Model to audit: **ConvNet (1.48 params)**
Audit data: **56 (48x48x1) images**

		EOD	SACC
Passive	2PC	1.62 hr	1.62 hr
	3PC	**29.92 sec**	29.84 sec
Active	2PC	11.873 hr	11.872 hr
	3PC	199.02 sec	198.99 sec

Times reported are time taken for MPC protocols to classify *and* compute fairness metrics. Evaluated on Google Cloud Platform (GCP) with 8 vCPUs, 32 GB RAM and egress bandwidth limited to 16 Gbps.

Future Directions

PrivFair can be

- Easily used and extended to report **other statistical notions** of fairness.
- Extended to **any ML model** for which secure inference is available.
- Extended to asynchronous computations where Bob can select pre-existing secret shares of the model to audit.

Acknowledgements Funding support for project activities provided by The University of Washington Tacoma's Founders Endowment fund, Canada CIFAR AI Chair, Facebook Research Award for Privacy Enhancing Technologies, and the Google Cloud Research Credits Program.