Bringing automation and fairness to identity verification on the internet with deep learning

Olivier Koch, VP of AI - Onfido

Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics

Special credits to **Yuanwei Li**, **Martins Bruveris**, and **Richard Tomsett**

Who are we?

Onfido is an online identity verification company.

We let our customers verify the identity of their users.

Current industries

- [x] Financial Services
- [x] Cryptocurrencies
- [x] Marketplaces
- [x] E-commerce
- [x] Transportation
- [x] Gaming
- [x] Healthcare

The future

- [ ] Hotels
- [ ] Airlines
- [ ] Telecoms
- [ ] Government

Onfido's 3 layers of identity verification

Do you have a **genuine** ID?

Are you a **real life** human?

Does your face **match** your ID?

1 — 2 — 3

onfido

Document Verification

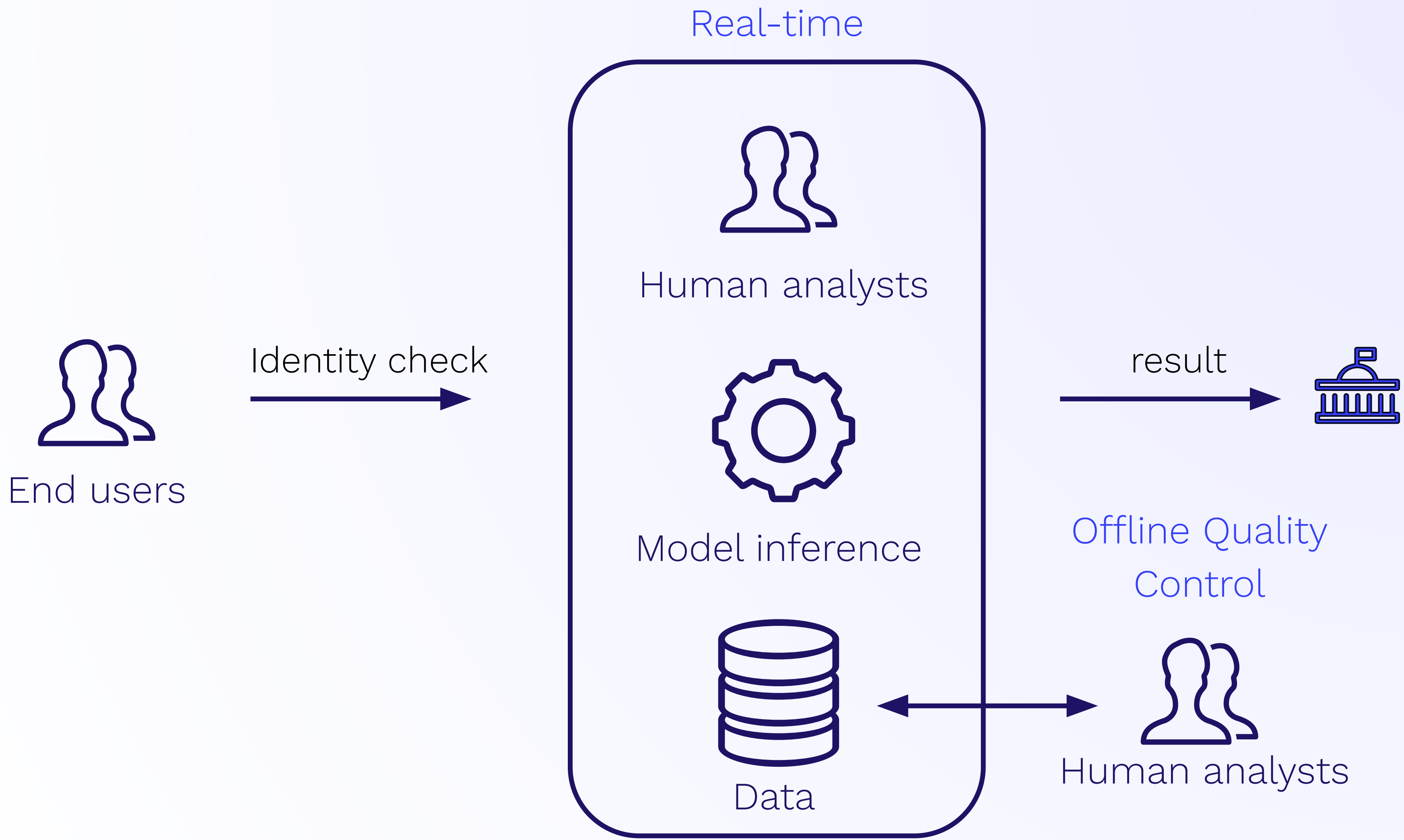
- Thousands of document types
- Constantly changing attack vectors
- Variable image quality (API vs SDK)
- Very low signal-to-noise ratio

Biometric Verification

+ Low friction and accessibility requirements
+ Bias reduction
+ Deepfakes and injection attacks

Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics


Real-time
Human analysts
Model inference
Data
End users
Identity check
result
Offline Quality Control
Human analysts

Automation brings several key benefits:

- Remove human variance
- More $ efficiency
- Better privacy
- Speed

At the cost of:

- More complexity (ML)
- AI risks (bias)

Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics

Let’s focus on document verification

Machine learning problem statement

Determine whether a document is fraudulent or not, given a large dataset of genuine samples and a (smaller) dataset of frauds

Across thousands of document types

Binary classification problem across many categories

Key metrics

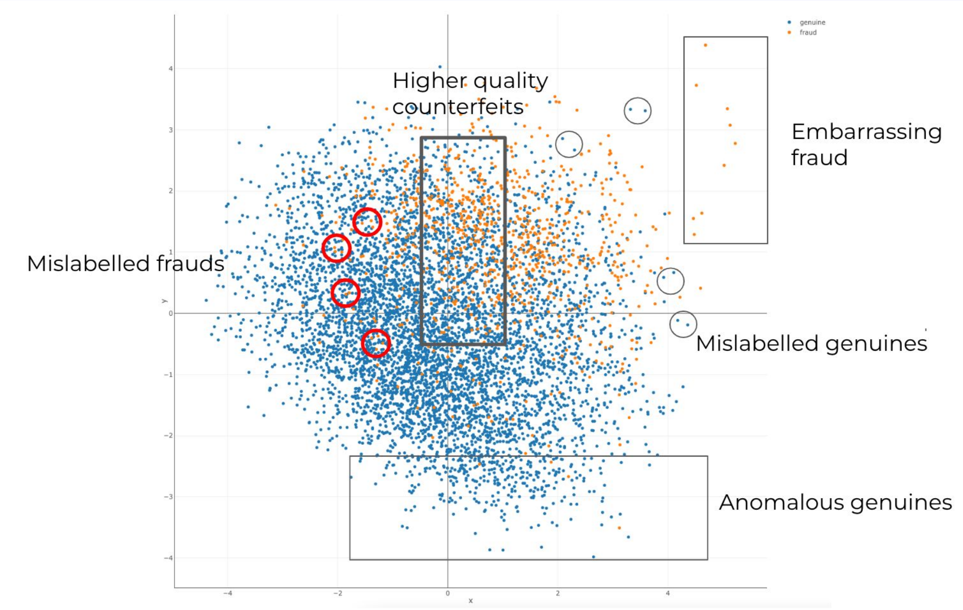
$$\text{FAR: False Acceptance Rate} = \frac{\#\ \text{accepted}}{\#\ \text{submitted}} \quad \text{(all frauds)}$$

$$\text{FRR: False Rejection Rate} = \frac{\#\ \text{rejected}}{\#\ \text{submitted}} \quad \text{(all genuine)}$$

Supervision beats unsupervised by a wide margin

Unsupervised (GMM): 5% FRR, 50% FAR

Supervised (LR, auto-encoders): 5% FRR, 10% FAR



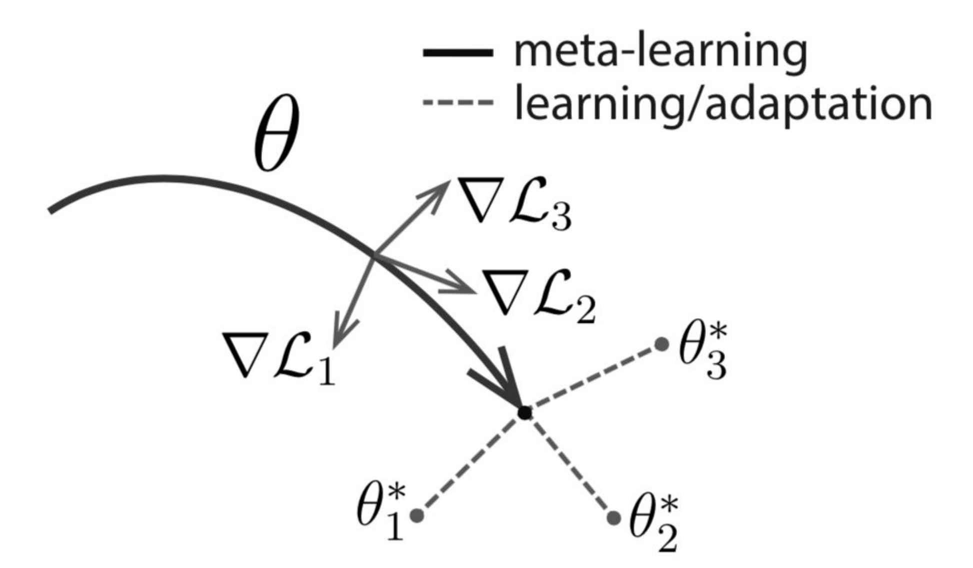
Three approaches

1. Train a single model for all doc types

1. Train a model per doc type

1. Meta-learning

Model-Agnostic Meta-Learning (Finn, et al. 2017)



Source: Meta Learning, learning to learn fast, Lilian Weng

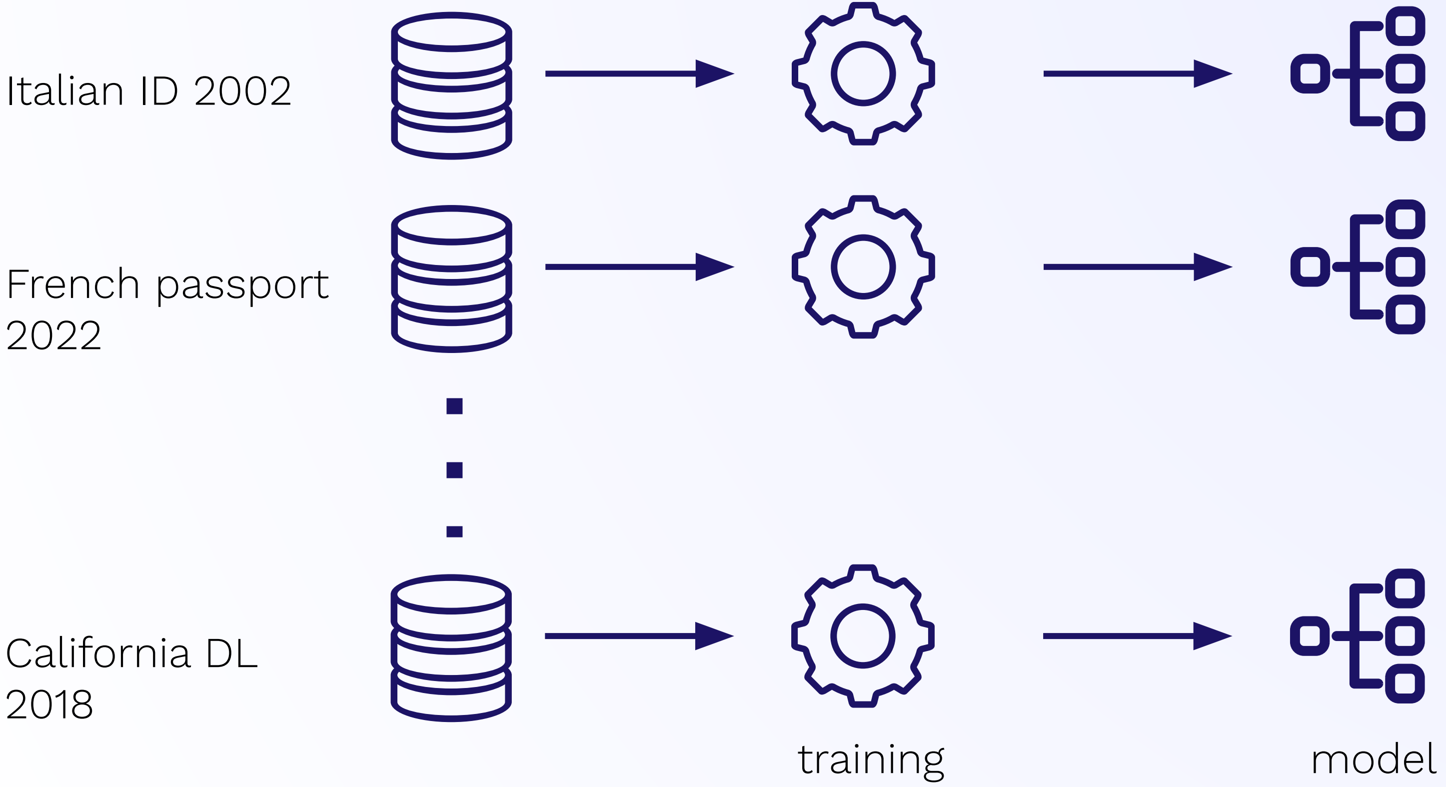
Model-Agnostic Meta-Learning (Finn, et al. 2017)

Algorithm 1 Model-Agnostic Meta-Learning

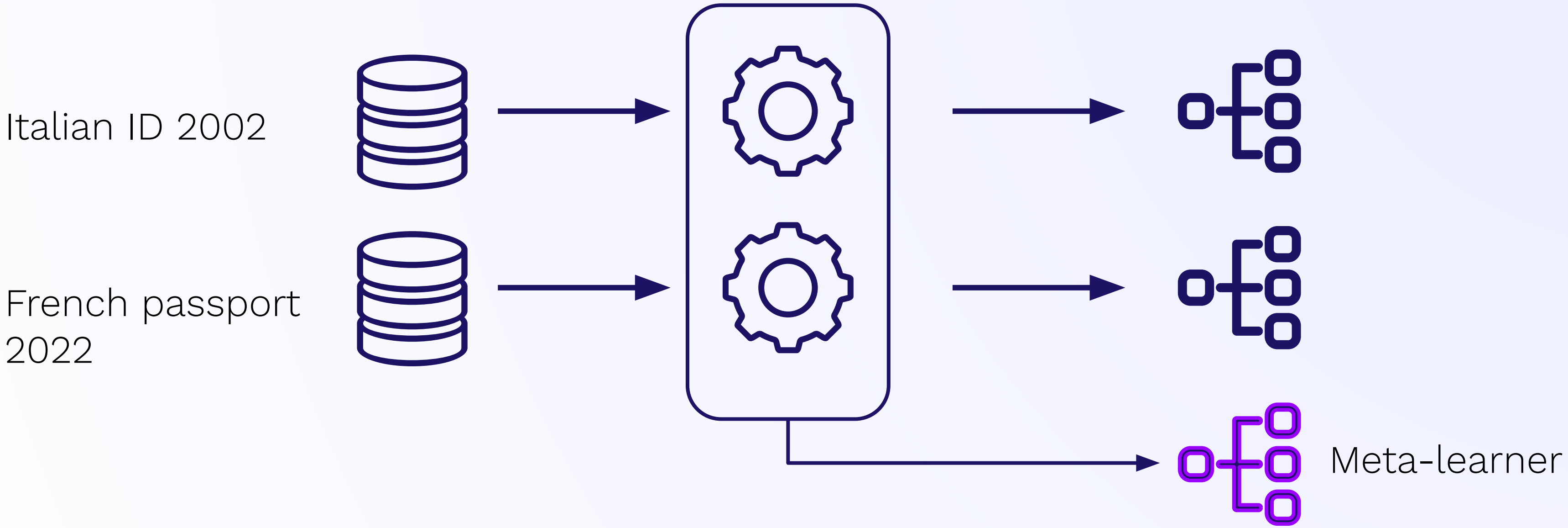
Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for** Note: the meta-update is using different set of data.
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

The general form of MAML algorithm. (Image source: original paper)

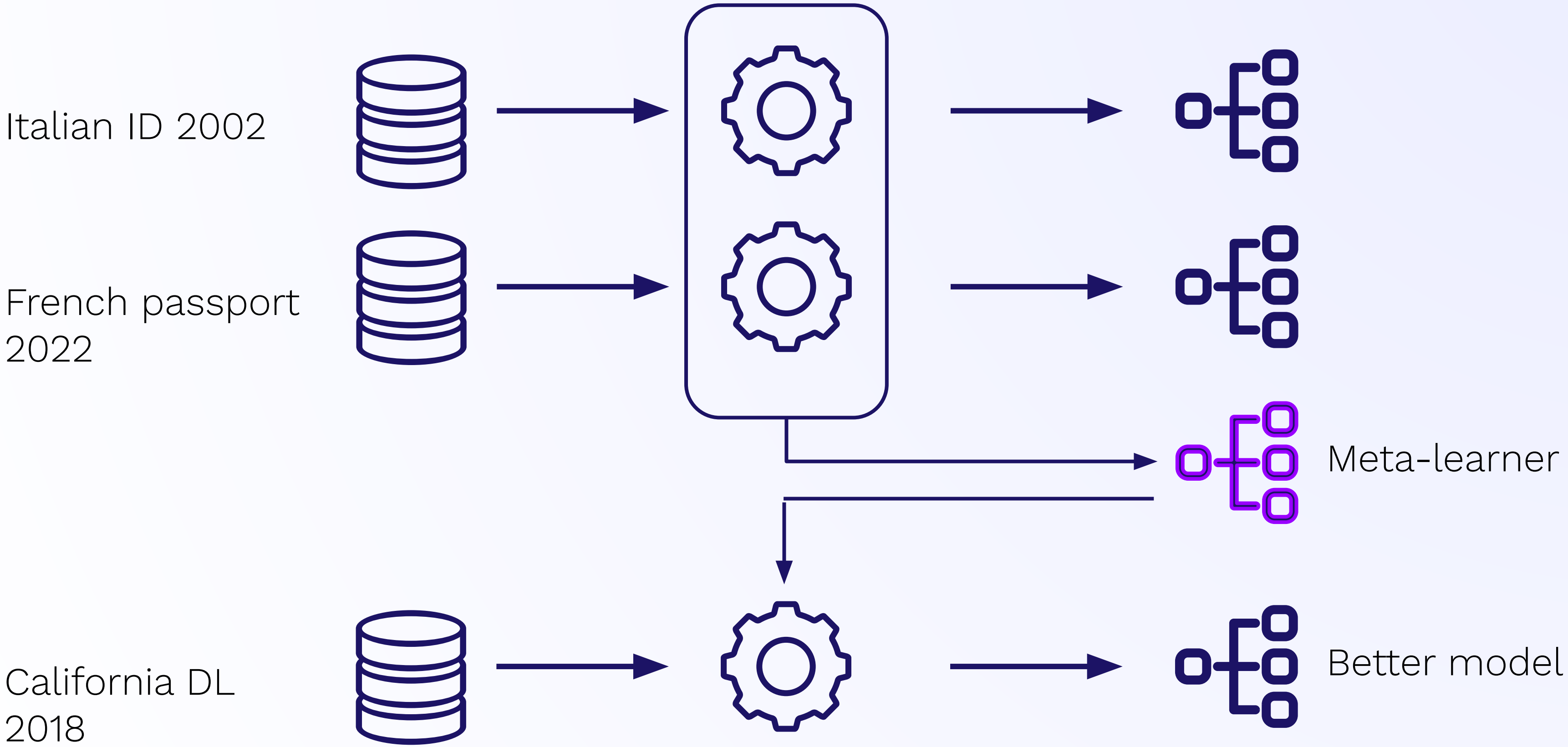
One model per document type



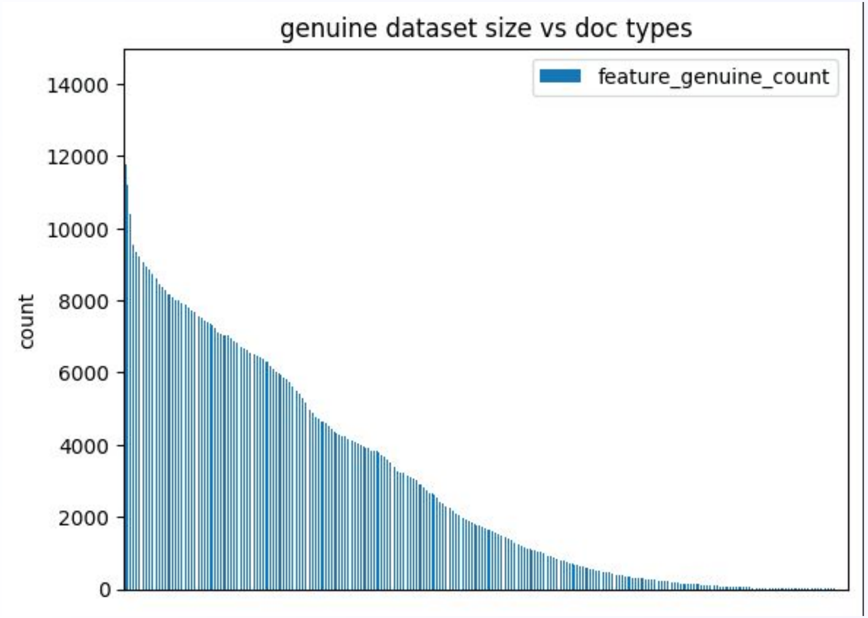
Meta-learning

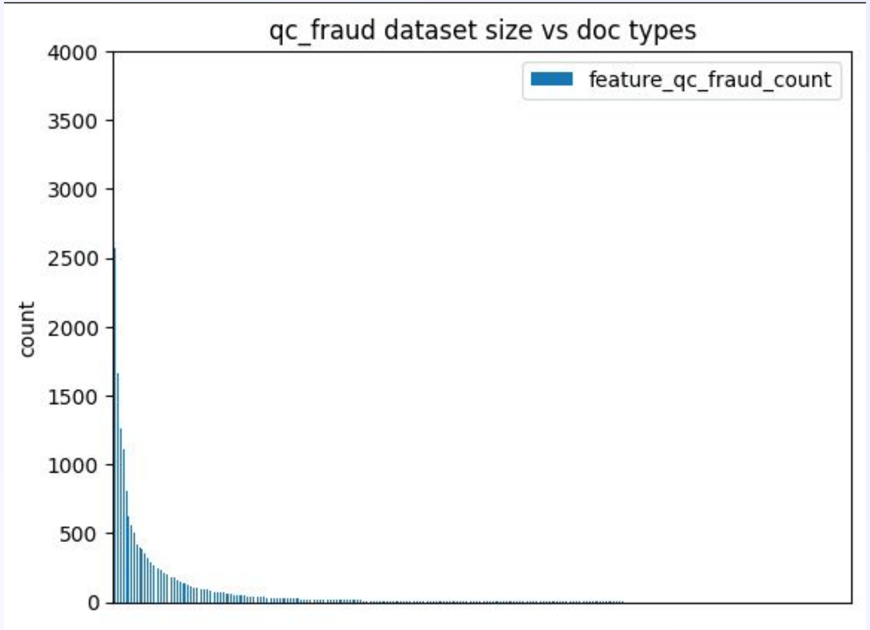


Meta-learning

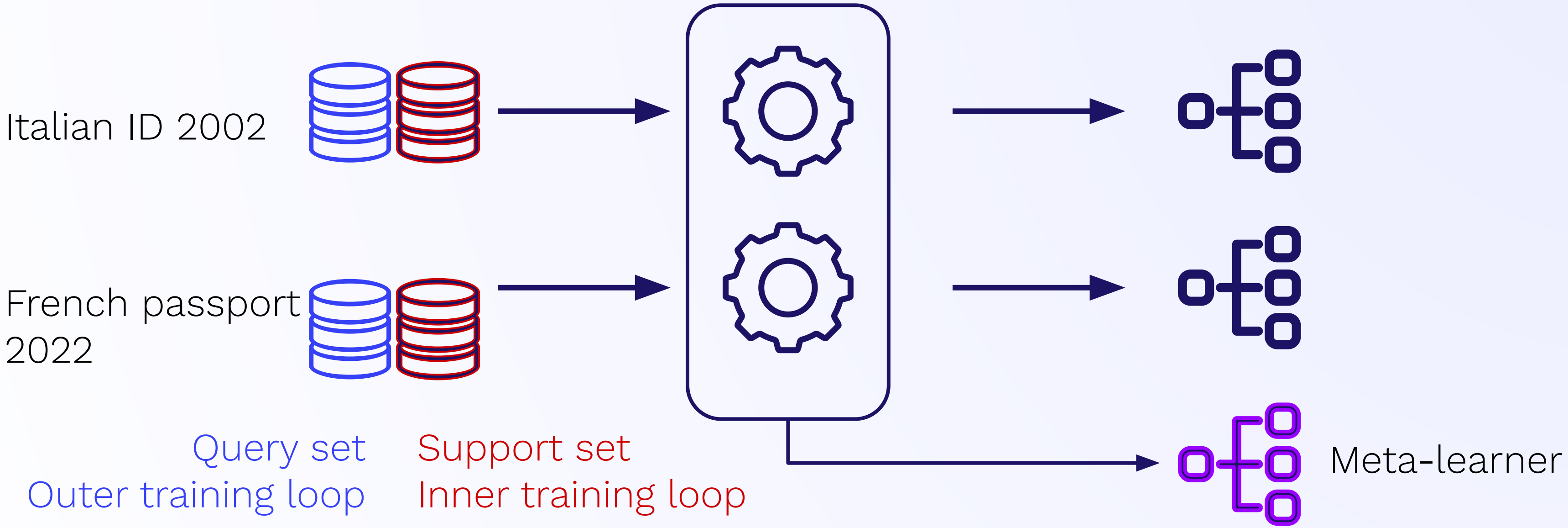


Particularly valuable for long-tail distributions



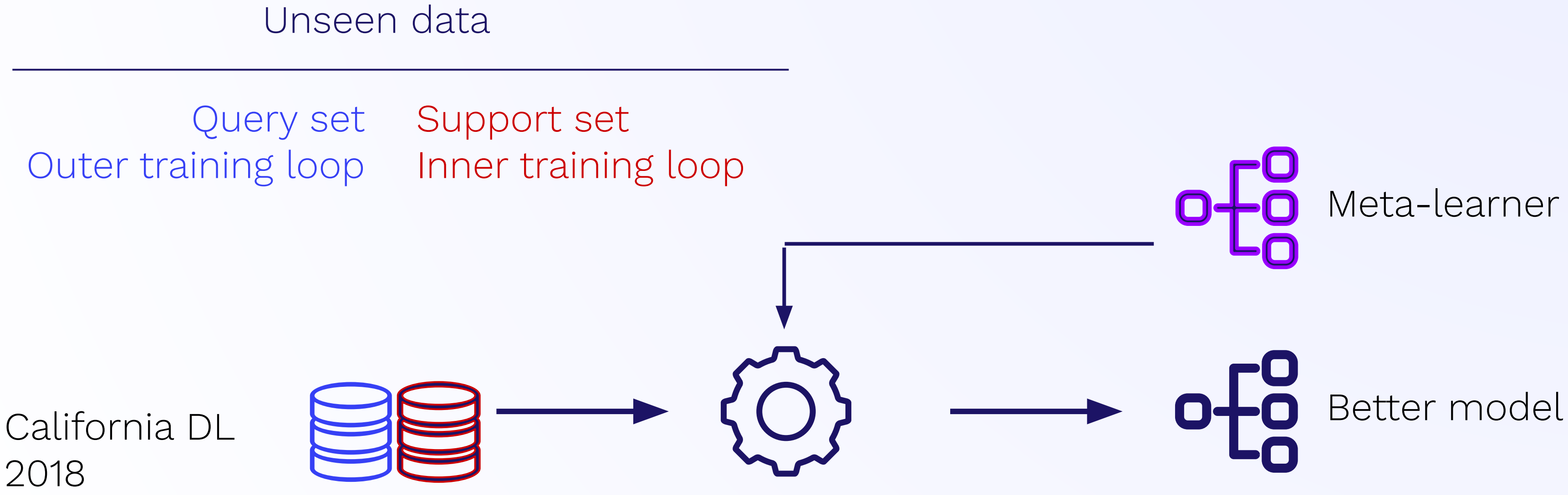


Meta-training



Each composed of K genuine
and K fraud samples

Meta-validation: train on support, evaluate on query



Validation split

- Split A: all data in training
- Split B: all top docs in validation
- Split C: a few top docs in validation

We present results on Split B

Experimental setup

Experiment	Setup
MAML 1	MAML with outer_lr=0.0001, inner_lr=0.1
MAML 2	MAML with outer_lr=0.0001, inner_lr=**2.0**
Pretrain	Supervised pre-training using MAML without inner loop. outer_lr=0.0001
Baseline	Random weight initialisation

We use the code from the original paper:

https://github.com/cbfinn/maml

Experimental setup (c'ed)

Fine-tuning method	Description
No fine-tuning (zero-shot inference)	The model weights from the training experiments are used directly for zero-shot inference without any fine-tuning on doc-specific training samples.
Fine-tune by steps	The model weights are fine-tuned on doc-specific training samples. We only use 1 genuine and 1 fraud samples for training. The performance is evaluated after a few steps (1,2,3,4,5,10) of model updates on the same pair of training examples.
Fine-tune by epochs	The model weights are fine-tuned on doc-specific training samples. We use a lot of genuines (thousands) and varying number of frauds for training. Fine-tuning is conducted for 60 epochs of the genuine data. Performance is evaluated when different number of training frauds are used.

Results on validation set with doc split B (21 docs held out)

Fine-tuning settings				FAR@FRR=0.02			
Fine-tune type	**# Train genuines**	**# Train frauds**	**Training duration**	**Baseline (random weights)**	**MAML 1**	**MAML 2**	**Pretrain**
No fine-tuning (zero-shot inference)	0	0	No training	0.9536	**0.3646**	0.6013	0.3824
Fine-tune by steps	1	1	1 step	0.9536	**0.3596**	0.3711	0.3824
	1	1	2 steps	0.9536	**0.3555**	0.4128	0.3825
	1	1	3 steps	0.9536	**0.3567**	0.3920	0.3827
	1	1	4 steps	0.9536	**0.3591**	0.3976	0.3826
	1	1	5 steps	0.9538	**0.3603**	0.3989	0.3823
	1	1	10 steps	0.9537	**0.3663**	0.4010	0.3814
Fine-tune by epochs	All	0	60 epochs	0.6337	0.5411	**0.5056**	0.5657
	All	1	60 epochs	0.5776	0.5013	**0.4555**	0.5008
	All	5	60 epochs	0.4587	0.4053	**0.3810**	0.4096
	All	10	60 epochs	0.3948	0.3520	**0.3283**	0.3618
	All	50	60 epochs	0.2352	0.2225	**0.2178**	0.2273
	All	100	60 epochs	0.2028	0.1923	**0.1919**	0.1953
	All	500	60 epochs	0.2019	0.1954	0.1972	**0.1947**
	All	1000	60 epochs	0.1576	0.1552	0.1565	**0.1510**

Results on validation set with doc split B (21 docs held out)

Fine-tuning settings				FAR@FRR=0.02			
Fine-tune type	**# Train genuines**	**# Train frauds**	**Training duration**	**Baseline (random weights)**	**MAML 1**	**MAML 2**	**Pretrain**
No fine-tuning (zero-shot inference)	0	0	No training	0.9536	0.364	0.6013	0.382
Fine-tune by steps	1	1	1 step	0.9536	**0.3596**	0.3711	0.3824
	1	1	2 steps	0.9536	**0.3555**	0.4128	0.3825
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3827
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	26
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	23
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4
Fine-tune b	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	57
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	08
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	96
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	18
	All	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.2273
	All	100	60 epochs	0.2028	0.1923	**0.1919**	0.1953
	All	500	60 epochs	0.2019	0.1954	0.1972	**0.1947**
	All	1000	60 epochs	0.1576	0.1552	0.1565	**0.1510**

MAML outperforms the best pretraining baseline on the zero-shot task (albeit by a small margin): 0.364 < 0.382

Results on validation set with doc split B (21 docs held out)

Fine-tuning settings				FAR@FRR=0.02			
Fine-t			tion	Baseline (random weights)	MAML 1	MAML 2	Pretrain
No fine-tun inference)				0.9536	**0.3646**	0.6013	0.3824
Fine-tune b				0.9536	**0.3596**	0.3711	0.3824
				0.9536	**0.3555**	0.4128	0.3825
				0.9536	**0.3567**	0.3920	0.3827
				0.9536	**0.3591**	0.3976	0.3826
				0.9538	**0.3603**		
				0.9537	**0.3663**		
Fine-tune by epochs	All	0	60 epochs	0.6337	0.5411	**0.5056**	0.5657
	All	1	60 epochs	0.5776	0.5013	**0.4555**	0.5008
	All	5	60 epochs	0.4587	0.4053	**0.3810**	0.4096
	All	10	60 epochs	0.3948	0.3520	**0.3283**	0.3618
	All	50	60 epochs	0.2352	0.2225	**0.2178**	0.2273
	All	100	60 epochs	0.2028	0.1923	**0.1919**	0.1953
	All	500	60 epochs	0.2019	0.1954	0.1972	**0.1947**
	All	1000	60 epochs	0.1576	0.1552	0.1565	**0.1510**

At the low fraud data regime, MAML outperforms pretraining significantly.

0.506

0.565

Results on validation set with doc split B (21 docs held out)

Fine-tuning settings				FAR@FRR=0.02			
Fine-tune type	**# Train genuines**	**# Train frauds**	**Training duration**	**Baseline (random weights)**	**MAML 1**	**MAML 2**	**Pretrain**
No fine-tuning (zero-shot inference)	0	0	No training	0.9536	**0.3646**	0.6013	0.3824
Fine-tune by steps	1	1	1 step	0.9536	**0.3596**	0.3711	0.3824
				0.9536	**0.3555**	0.4128	0.3825
				0.9536	**0.3567**	0.3920	0.3827
				0.9536	**0.3591**	0.3976	0.3826
				0.9538	**0.3603**	0.3989	0.3823
				0.9537	**0.3663**	0.4010	0.3814
				0.6337	0.5411	**0.5056**	0.5657
				0.5776	0.5013	**0.4555**	0.5008
				0.4587	0.4053	**0.3810**	0.4096
	All	10	60 epochs	0.3948	0.3520	**0.3283**	0.3618
	All	50	60 epochs	0.2352	0.2225	**0.2178**	0.2273
	All	100	60 epochs	0.2028	0.1923	**0.1919**	0.1953
	All	500	60 epochs	0.2019	0.1934	0.1972	**0.1947**
	All	1000	60 epochs	0.1576	0.1552	0.1565	**0.1510**

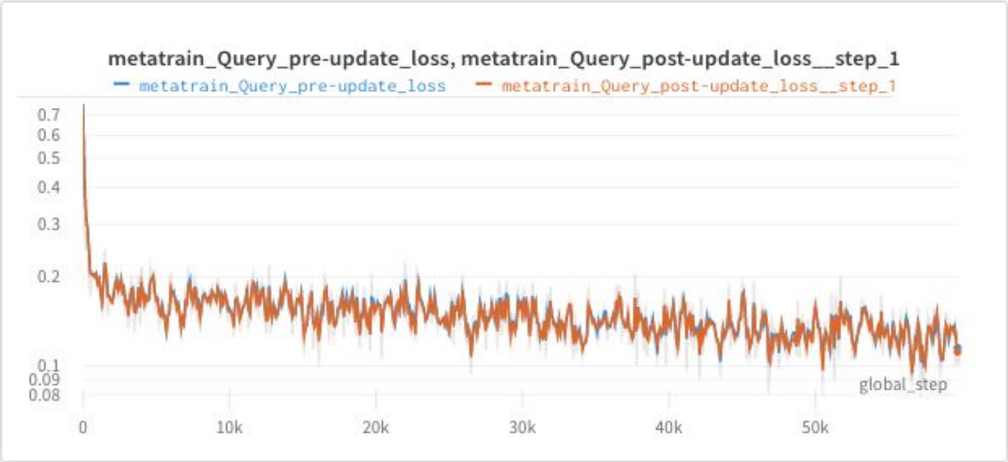
At the high fraud data regime, all methods are on par.

Results on validation set with doc split B (21 docs held out)

Fine-tuning settings				FAR@FRR=0.02			
Fine-tune type	**# Train genuines**	**# Train frauds**	**Training duration**	**Baseline (random weights)**	**MAML 1**	**MAML 2**	**Pretrain**
No fine-tuning (zero-shot inference)	0	0	No training	0.9536	**0.3646**	0.6013	0.3824
Fine-tune by steps	1	1	1 step	0.9536	**0.3596**	[illegible]	824
	1	1	2 steps	0.9536	**0.3555**	[illegible]	825
	1	1	3 steps	0.9536	**0.3567**	[illegible]	827
	1	1	4 steps	0.9536	**0.3591**	[illegible]	826
	1	1	5 steps	0.9538	**0.3603**	[illegible]	823
	1	1	10 steps	0.9537	**0.3663**	[illegible]	814
Fine-tune by epochs	All	0	60 epochs	0.6337	0.5411	[illegible]	657
	All	1	60 epochs	0.5776	0.5013	[illegible]	008
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.405	[illegible]	096
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.3520	[illegible]	618
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.2225	[illegible]	273
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.1923	[illegible]	953
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.1954	[illegible]	**947**
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.1552	0.1565	**0.1510**

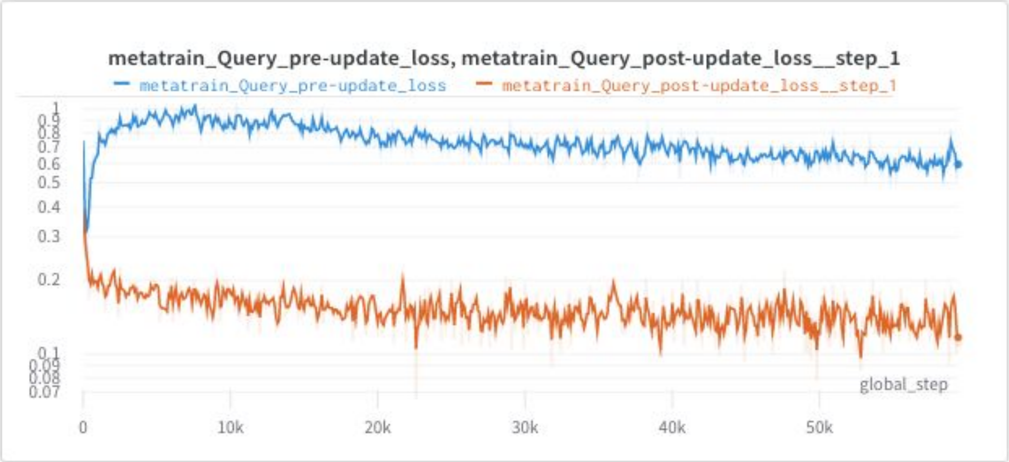
Fine-tuning on a single sample, the best performance is reached with the same number of training steps that was used during training (1 step).

Zooming in on the outer loop training (pre-update loss, post-update loss)



MAML1: inner loop learning rate too small (lr = 0.1)

MAML2: inner loop is working (lr = 2.0)



Our results support a “feature reuse” scenario

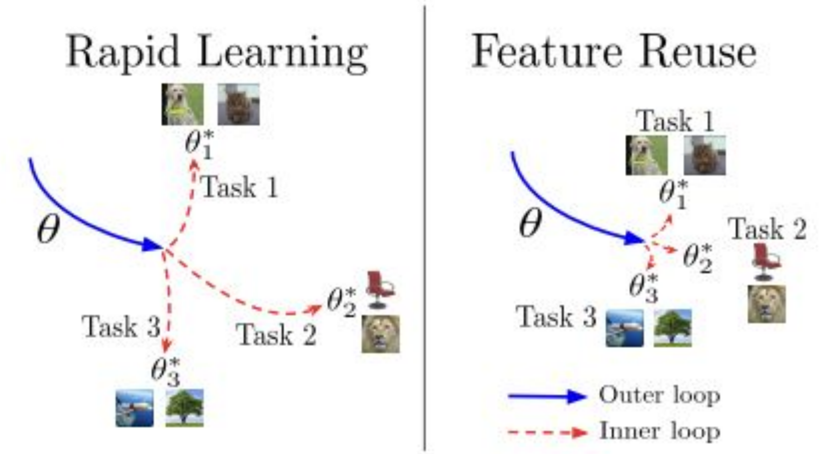


Figure 1: **Rapid learning and feature reuse paradigms.** In Rapid Learning, outer loop training leads to a parameter setting that is well-conditioned for fast learning, and inner loop updates result in significant task specialization. In Feature Reuse, the outer loop leads to parameter values corresponding to reusable features, from which the parameters do not move significantly in the inner loop.

Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML, ICLR 2020

MAML allows to get the best of both worlds:

- Best performance in low-data regime
- On-par with pretraining in high-data regime

Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics

Several definitions of bias

Demographic parity

Equality of opportunity

Equality of odds

Predictive parity

[Source: Google glossary on fairness]

Equality of opportunity

Candidates are equally likely to be admitted irrespective of which group they belong to, as long as they are qualified.

Equality of opportunity in supervised learning, Hardt, Price and Srebro, NeurIPS, 2016

Proposed metric for fairness in identity verification

FRR should be the same across groups.

Measure FRR/group and normalize by overall FRR.

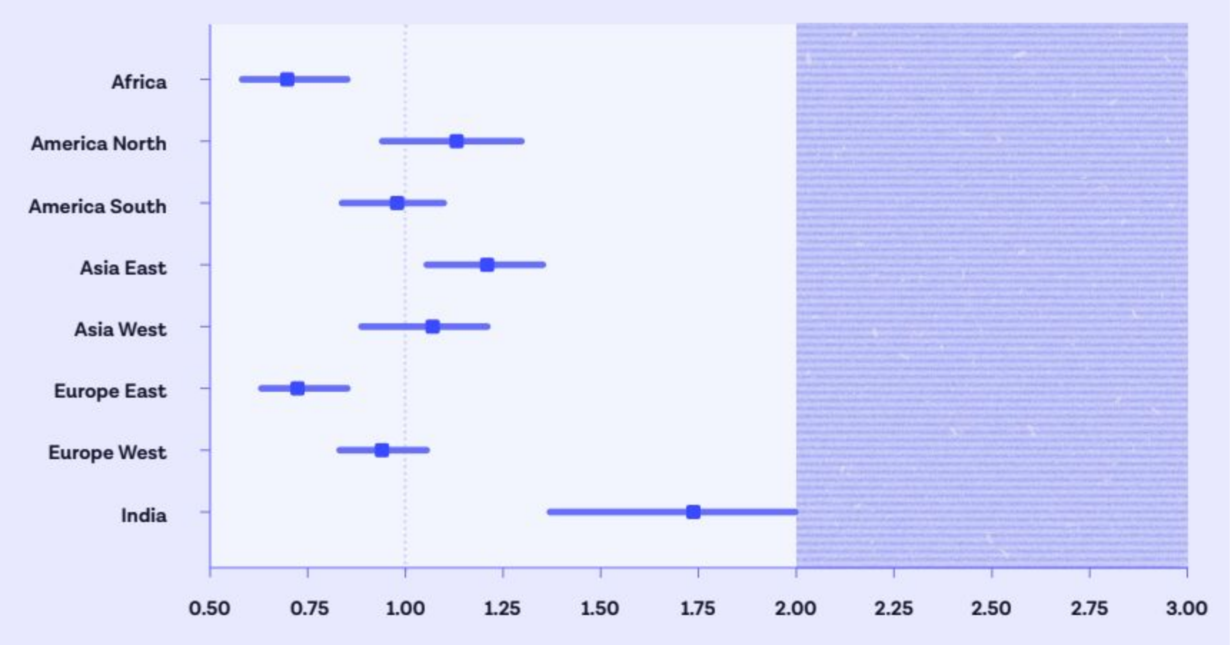
Ratio > 1: group is over-rejected

Ratio < 1: group is under-rejected

Bias mitigation: demographic differential for Motion

Source: “Building without bias”, Onfido

FRR bias against overall population (1.0 = no bias)



95% confidence intervals

Bias mitigation: demographic differential for Motion

From our latest white paper “Building without bias”

Gender bias

In regard to gender we observe some bias between male or female, with a ratio of 0.87 for male and 1.18 for female.

	Male	Female
Group FRR / Overall FRR (95% confidence interval)	**0.87** (0.82 - 0.92)	**1.18** (1.11 - 1.26)

Bias mitigation: demographic differential for Motion

From our latest white paper “Building without bias”

Age bias

In regard to age groups we see a tight grouping of ratios in all but the over 50 group.

	<25	25-30	30-40	40-50	>50
Group FRR / Overall FRR	**0.89**	**0.83**	**0.87**	**1.24**	**1.71**
(95% confidence interval)	(0.81 - 0.96)	(0.76 - 0.93)	(0.80 - 0.95)	(1.07 - 1.42)	(1.51 - 1.95)

Reducing bias, practical considerations

Modify the dataset

Change the training procedure

Apply post-processing to the output of the model

Conclusion

Identity verification is a core function of our digital lives

Automating identity verification brings many benefits

Meta-learning > supervised learning >> unsupervised

Bias matters and we propose a pragmatic approach to it

Future areas of research

Better meta-learning models

Self-supervised learning

Generative models for realistic synthetic data