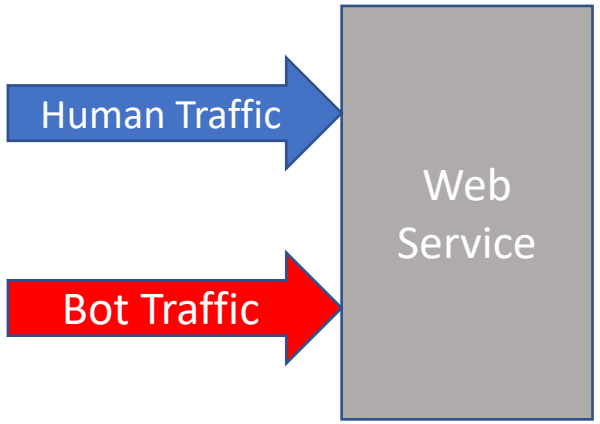
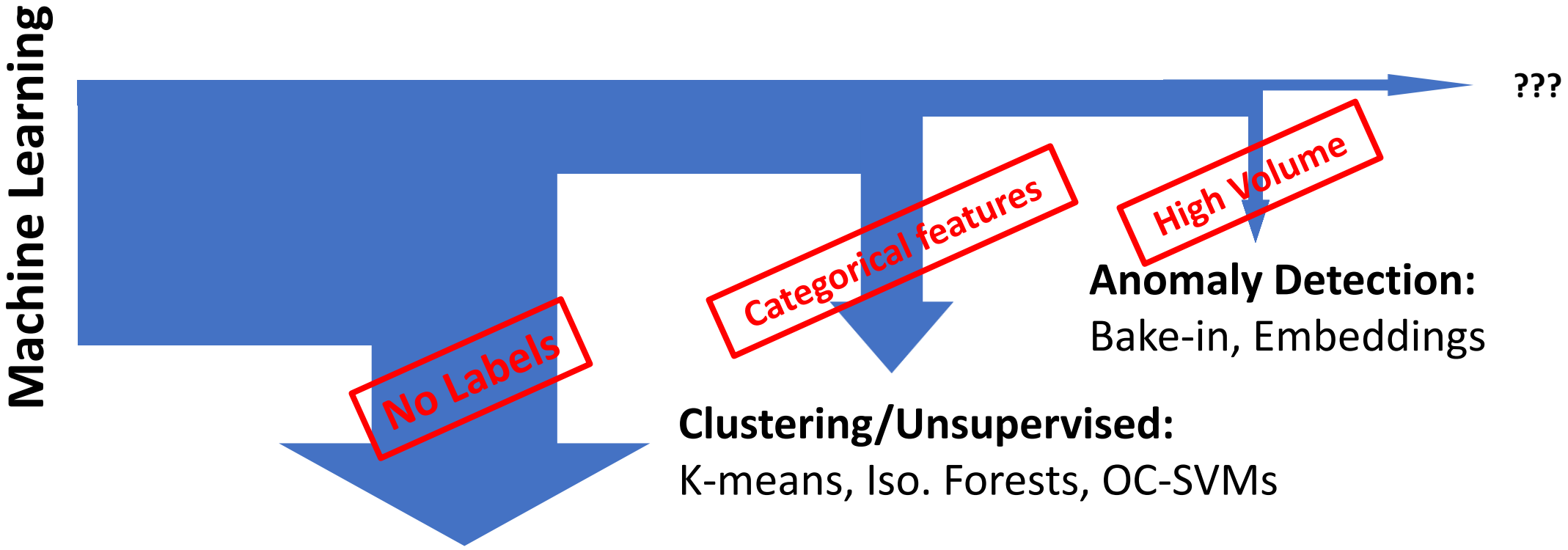
Automated Detection of Automated Traffic

Cormac Herley, Microsoft



Examples: Signup abuse, Password guessing, CAPTCHA solving, click-fraud, Inauthentic engagement, …….

No labels, Categorical features, high volume abuse



Punish deviations from Clean

$$P(x,y) = \alpha\, P(x,y|cln) + (1-\alpha)\, P(x,y|bot)$$

If we knew $P(x,y|cln)$ then odds of being malicious:

$$Odds = \frac{P(Bot|x,y)}{P(Cln|x,y)} = \frac{P(x,y)}{\alpha\, P(x,y|cln)} - 1$$

Unattacked bins of y → Clean dist. of x

$$P(x,y) = \alpha\, P(x,y|cln) + (1-\alpha)\, P(x,y|bot)$$

Assumptions:

- *x, y independent:* $P(x,y|cln) = P(x|cln)\, P(y|cln)$
- $\exists$ *unattacked bin* y_j

$$P(x,y_j) = \alpha\, P(x|cln)\, P(y_j|cln) + (1-\alpha)\, P(x,y_j|bot)$$

$P(x,y_j|bot) \to 0$

$$\Rightarrow P(x|cln) = \text{const.}\ P(x|y_j)$$

1. *Why would there be unattacked bins?*

2. *How do we find them?*

1. *Multiple unattacked bins give same distribution*
2. *Multiple bins giving same distribution are unattacked**

If we find a cluster of k equal marginals P(browser|state):

$$P(browser|Iowa) = P(hour|Oregon) = P(browser|Georgia) = P(browser|cln)$$

Assuming P(state|cln) ≠ P(state|bot)

*Unless attacker can match Clean(y,z) over subspace of dimension #bins

Joint distribution matrix $\Omega = P(x)P(y)^T$

$$\Omega = \alpha \cdot \Sigma + (1-\alpha) \cdot \Theta$$

Full Rank

Rank-1 since $x \perp y$

$P(x|cln) = Span\{Rank\text{-}1 \text{ subset of cols of } \Omega\}$

```
def getCleans():
    for f in {w, x, y, z}:
        pivot(idx = f, cols = [features indep of f], aggfunc = sum())
        Cluster over cols          # use history.Clean(f) for regularization)

def getRules()
    for (w, x, y, z) in {w, x, y, z}.unique():
        Odds(w,x,y,z) = Observed(w,x,y,z)/ (α C(w)C(x)C(y)C(z)) -1
```

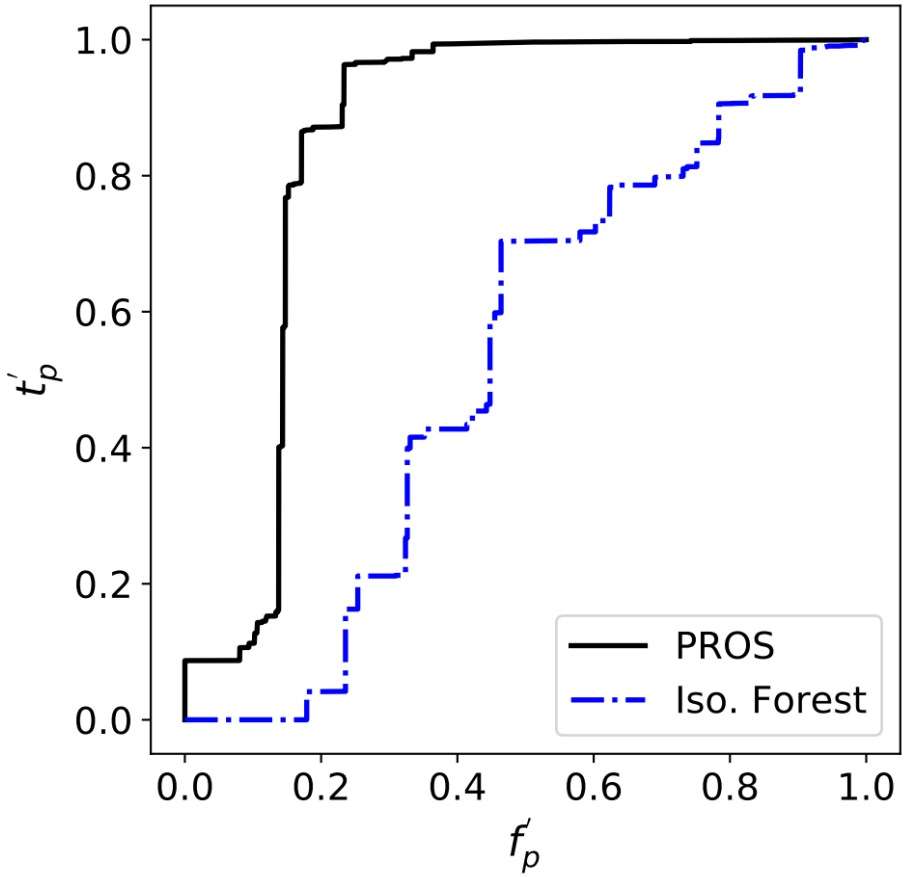
country	state	browser	org	Odds
us	ca	Samsung Internet12.1	cox communications inc.	0.155896
us	va	Edge84.0.522	charter communications	-0.845446
us	ny	Chrome40.0.2214	stingers inc.	6532.995753

Evaluation

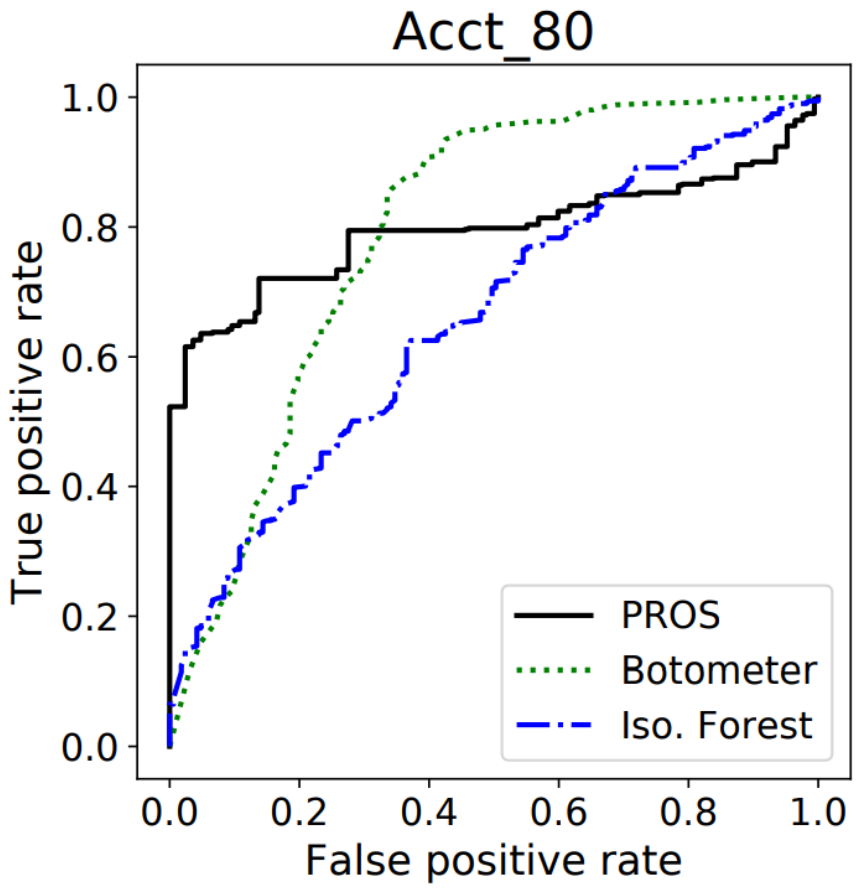
Open-source web-server logs: secrepo.com

- Partial labels:
  - Requests refused by site firewall
  - Requests for non-existent Wordpress admin and logins

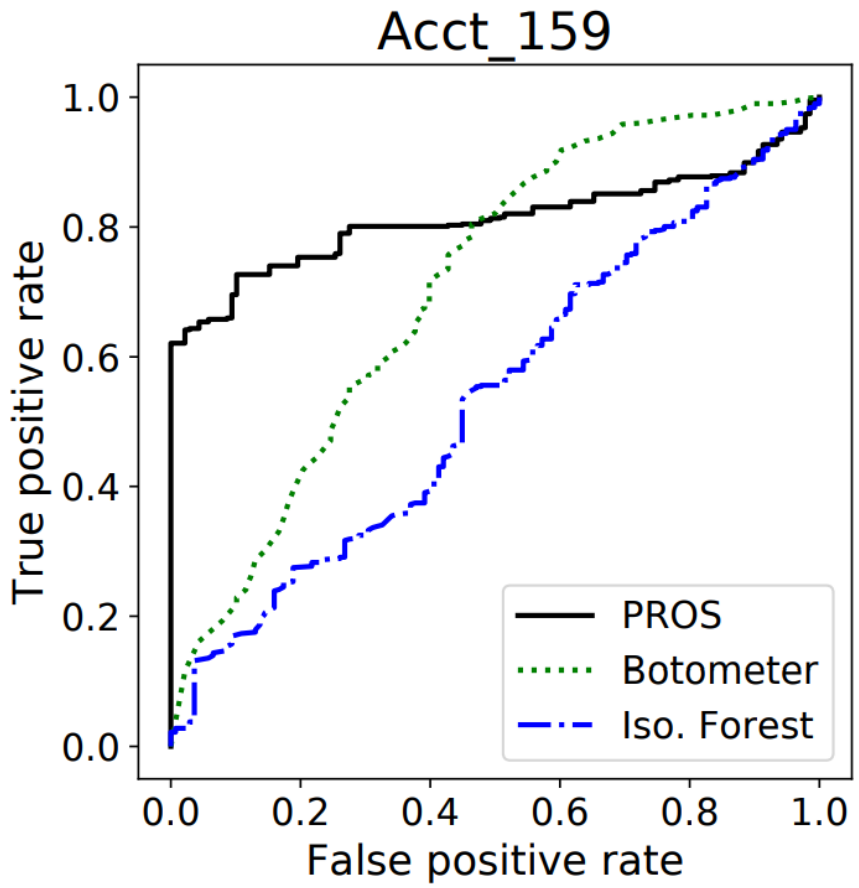
- $t_p \geq t'_p$ and $f_p \leq f'_p$
- Actual ROC above, left of shown
- Features = [browser, family, status, path]
- AUC = 0.877, Iso. Forest=0.532



Twitter: 2 accts w/ known bot followers



AUC: 0.795, 0.792, 0.657



AUC: 0.811, 0.708, 0.531

Features = [account, username pattern, client, year]

Crawl of 5.1m Twitter Accts

Target	Pattern	Client	Year	Count	Odds	Btmtr
Acct_8	ldd	Mobile Web (M2)	2013	404	692.85	0.93
Acct_8	other	Mobile Web (M2)	2013	381	391.03	0.88
Acct_74	l	Mobile Web	2009	189	165.61	0.81
Acct_74	other	Mobile Web	2009	110	110.31	0.86
Acct_74	l	Twitter Web Cl.	2009	3383	100.40	0.79
Acct_12	Ul	None	2012	1208	73.14	0.81
Acct_8	other	-1	2012	625	72.56	0.84
Acct_46	Uld	None	2016	842	68.57	0.88
Acct_74	ldd	Twitter Web Cl.	2009	547	64.17	0.80
Acct_46	UlUld	None	2016	1512	63.86	0.82

E.g., Cluster of 404:
Lastname+2letter+2digits

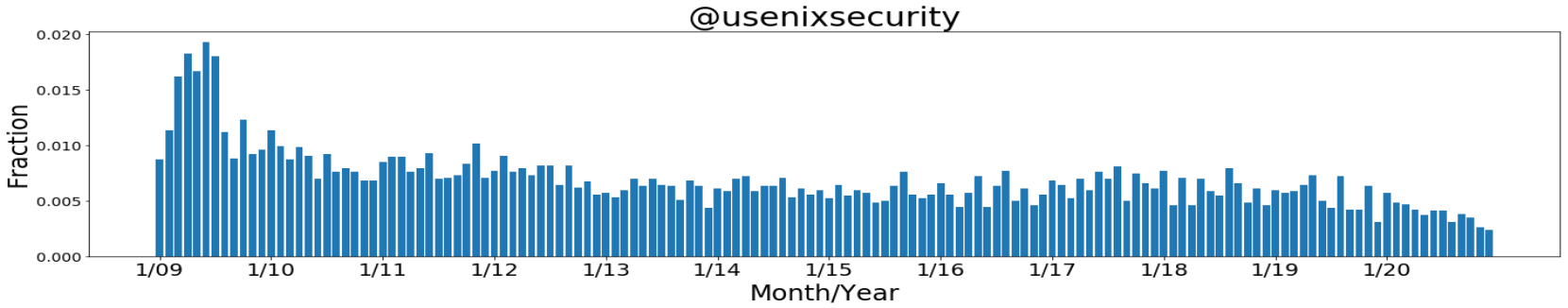
Colwellun33
Niceleson ot49
Fischbachqx34

⋮

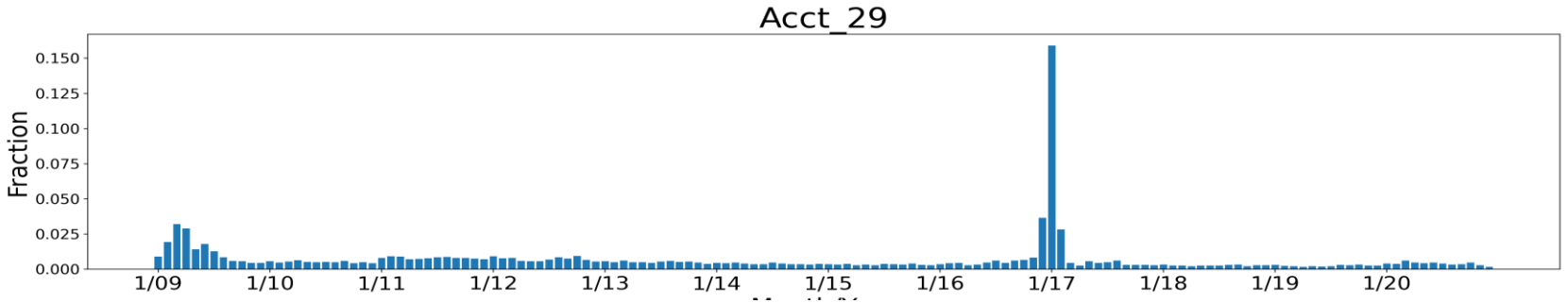
Langstonka76
Whistlerps42
Johnsonaq04

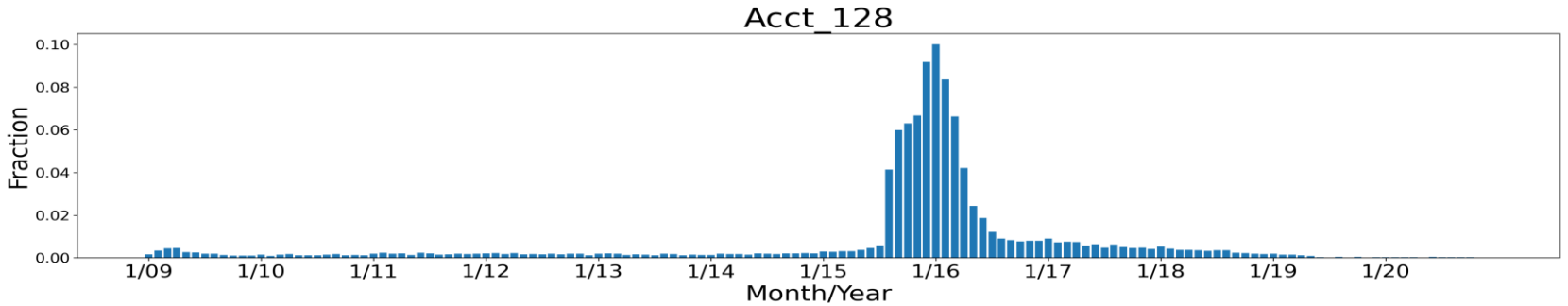
Crawl of 5.1m Twitter Accts

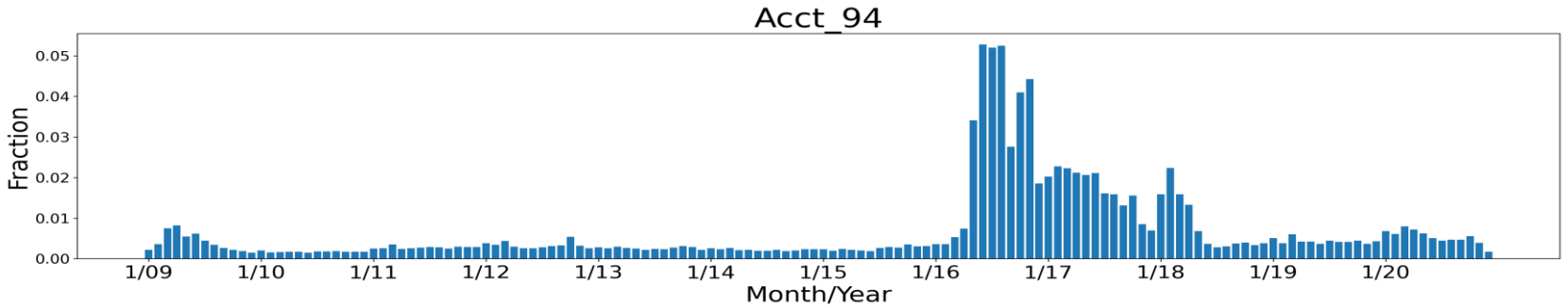
(≈50% related to 2020 Election)



approx. Clean







Conclusions

When x, y independent

P(x|cln) = Span{Rank-1 subset of cols of P(x,y)}

- ***Unsupervised***
- ***Categorical features***
- ***One-sided:***
  - *not assuming* $P(x|Bot, Train) \approx P(x|Bot, Deploy)$