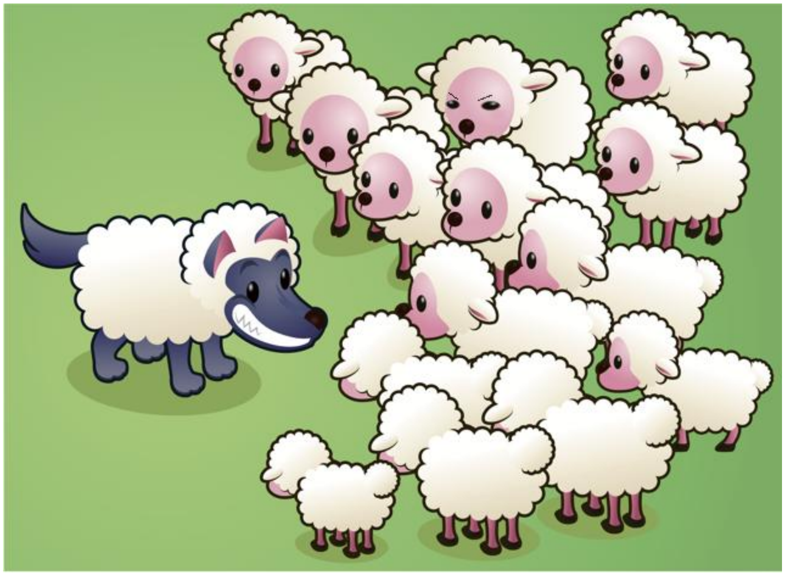
Machine Learning Adversarial Label Tampering: Design and Detection



IF (white AND fuzzy) THEN <Harmless>

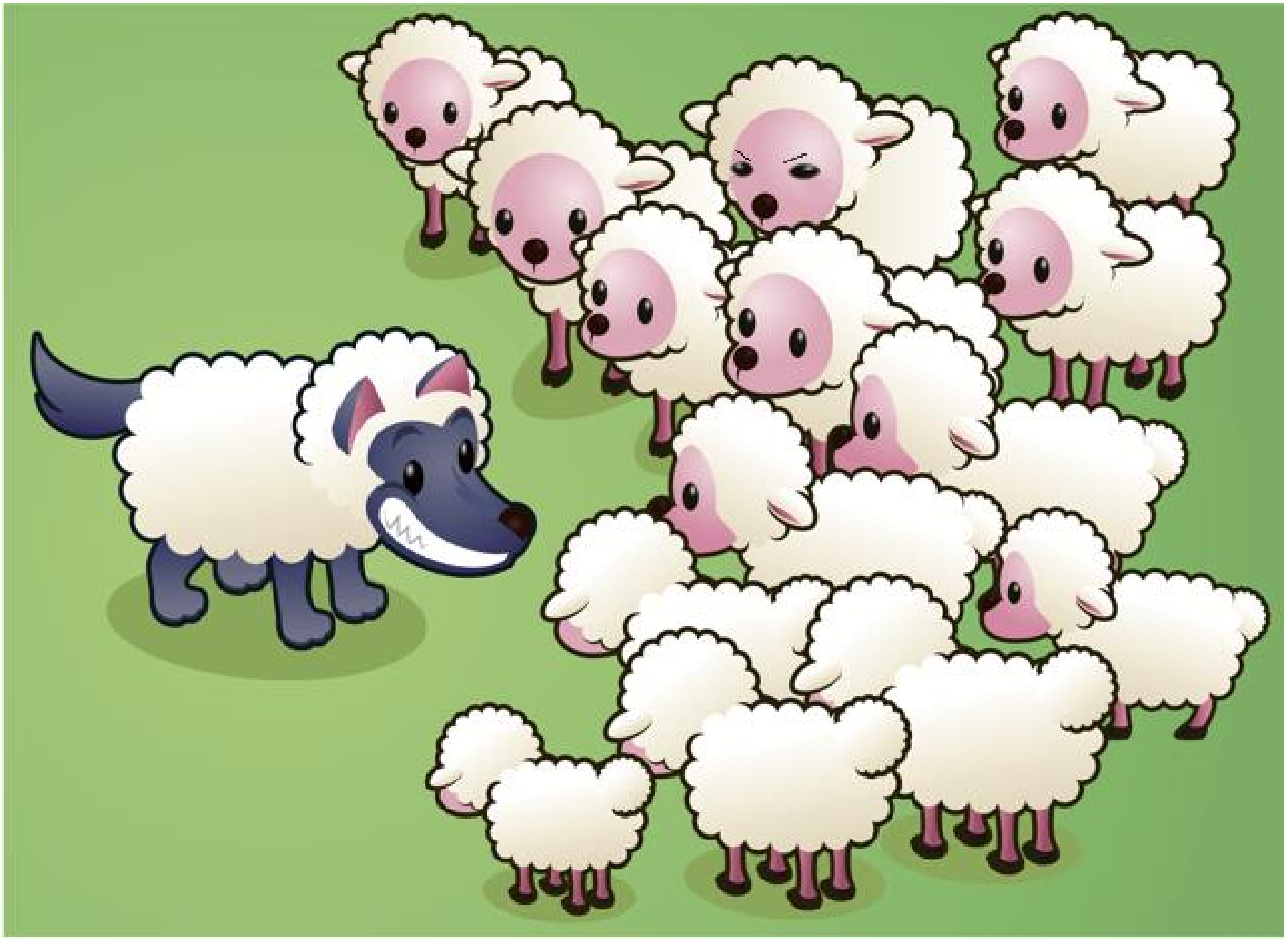
Philip Kegelmeyer

Sandia National Laboratories, Livermore, CA

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Computational Cybersecurity in Compromised Environments, September 18, 2018

Counter Adversarial Data Analytics (CADA)



IF (white AND fuzzy) Then <Harmless>

Philosophy

We must learn to love life . . .

. . . without ever trusting it[3].

⇒ “We must learn to love ~~life~~ data . . . without ever trusting it.”

The broad question: how to turn this into quantifiable, practical advice?

Outline

- **“Adversarial”** ambition is ambiguous (and alliterative).
- Machine learning has default expectations.
- These are deceptively subverted by label tampering attacks.
- There are a variety of possible label tampering attacks.
- “Quantified paranoia” might be one way to detect them.

"Adversarial" Ambition is Ambiguous

The word "adversarial" has many distinct connotations.

An incomplete list of possible adversarial goals and models:

A) Undermine the sensor

B) Model stealing

C) Generative adversarial networks

D) Test sample attacks on deep learning image analysis

E) An *algorithmically* informed, empowered adversary

A) Undermine a Sensor

Classic adversarial methods

Jamming

Hiding

Deception[4]

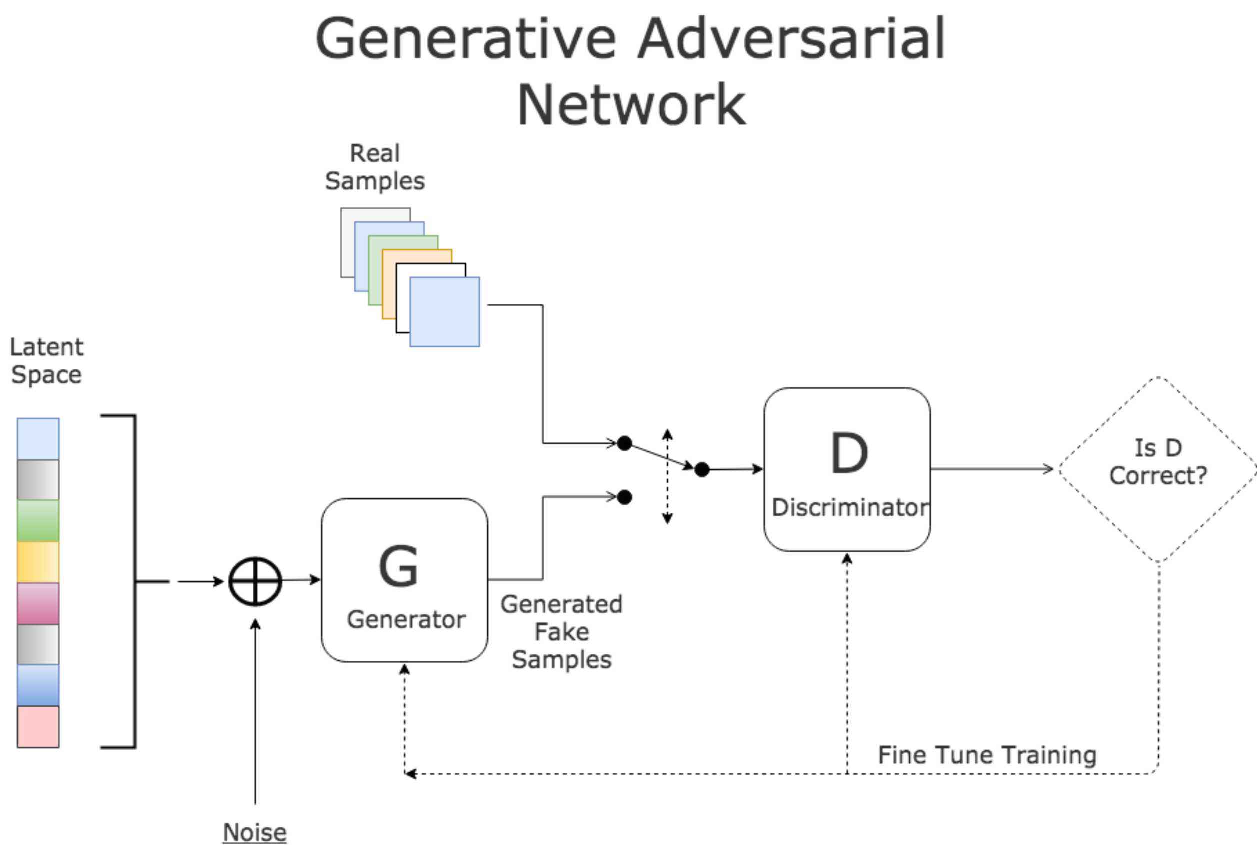
B) Model Stealing

An adversary who copies or reverse engineers a machine learning model, likely in order to study it and build custom attacks against it[10, 9].

Credit: Dooder, Freepik.com

C) Generative Adversarial Networks

More like resistance training[6] than malevolent adversarial action.



From KDDNuggets, January 2017

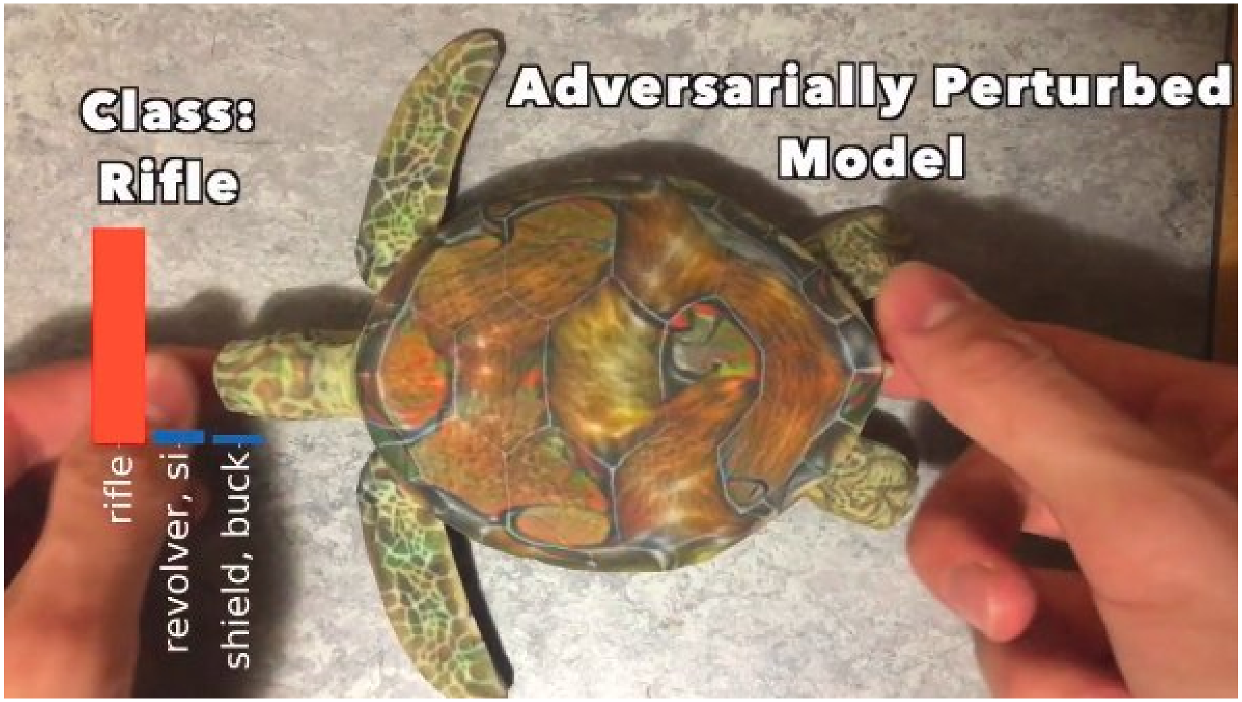
D) Test Sample Attacks on DL Image Analysis

Many recent examples:

Robust Physical-World Attacks on Machine Learning Models

Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples

Fooling Neural Networks in the Physical World with 3D Adversarial Objects ...



Synthesizing Robust Adversarial Examples, Anish Athalye, Logan Engstrom, Andrew Ilyas, Kevin Kwok

E) An *Algorithmically* Informed Adversary

A worst case scenario: an adversary that knows every detail of our machine learning method, *and* has some ability to alter the data.

We aim to quantify just how badly we are hosed.

E) An *Algorithmically* Informed Adversary

A worst case scenario: an adversary that knows every detail of our machine learning method, *and* has some ability to alter the data . . .

Recent papers to know if you use deep learning with pre-trained networks:

- *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*, Tianyu Gu, Brendan Dolan-Gavitt, Siddharth Garg
- *Machine Learning Models that Remember Too Much*, Congzheng Song, Thomas Ristenpart, Vitaly Shmatikov

Outline

- “Adversarial” ambition is ambiguous (and alliterative).
- **Machine learning** has default expectations.
- These are deceptively subverted by label tampering attacks.
- There are a variety of possible label tampering attacks.
- “Quantified paranoia” might be one way to detect them.

Machine Learning In One Slide

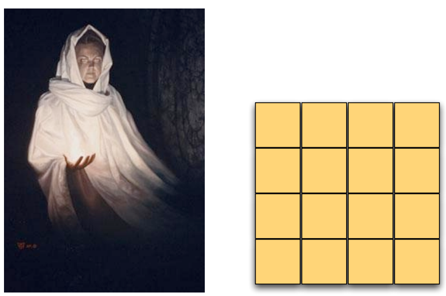
id	Truth	a_1	a_2	a_3	$\ldots$	a_K
q_0	I	8	612	0.57	$\ldots$	0.70
q_1	R	12	1003	0.97	$\ldots$	0.12
q_2	R	99	2	0.33	$\ldots$	0.03
q_3	I	3	27	0.12	$\ldots$	0.13
q_4	R	16	183	0.08	$\ldots$	0.58
q_5	I	17	665	0.36	$\ldots$	0.64
q_6	I	44	1212	0.29	$\ldots$	0.42
q_7	I	42	24	0.33	$\ldots$	0.88
q_8	R	78	42	0.44	$\ldots$	0.52
q_9	I	32	111	0.83	$\ldots$	0.71

(Of course, no real training set would have just ten samples.)

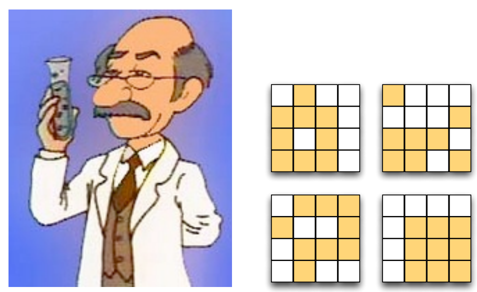
Ensemble Machine Learning In One Slide

Start with "ground truth" training data:
each training sample has attributes and *trusted* labels.

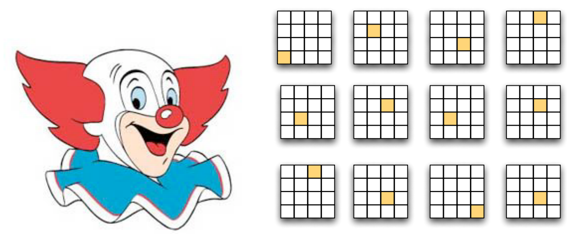
Sage sees all the data.



Experts see diverse subsets.



Each bozo sees a tiny fraction.



The experts beat the sage[1]. The bozos beat the experts[2].

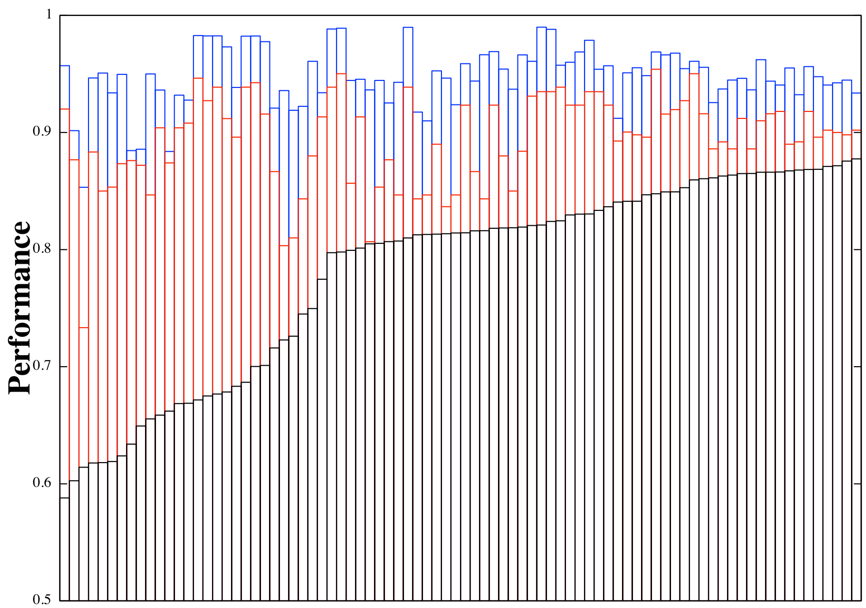
Outline

- “Adversarial” ambition is ambiguous (and alliterative).
- Machine learning has **default expectations**.
- These are deceptively subverted by label tampering attacks.
- There are a variety of possible label tampering attacks.
- “Quantified paranoia” might be one way to detect them.

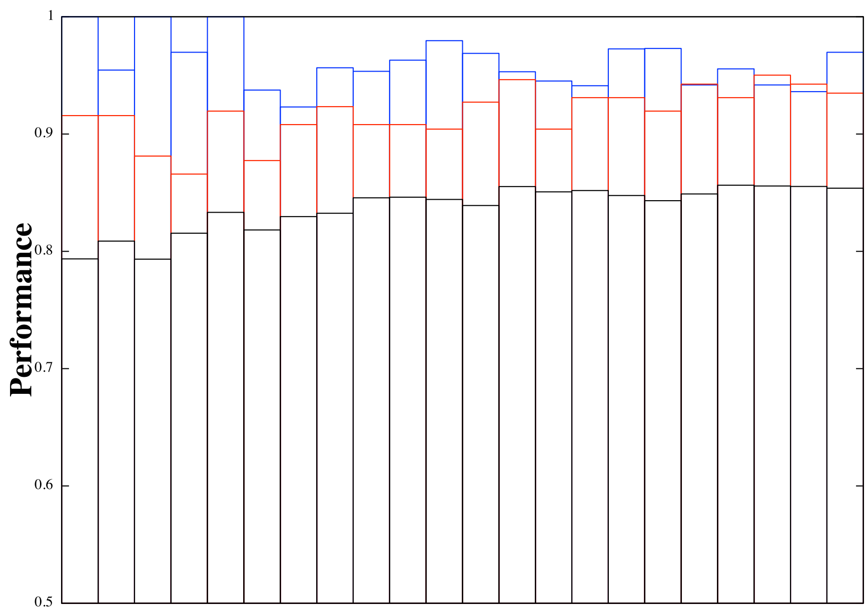
Review: Performance Assessment Expectations

Typically, one expects:

- **cross-validation on the training data** to be an optimistic estimate . . .
- . . . of **ensemble performance on test data**, which in turn is better than
- . . . **non-ensemble performance on test data**.



Various Algorithm Parameters

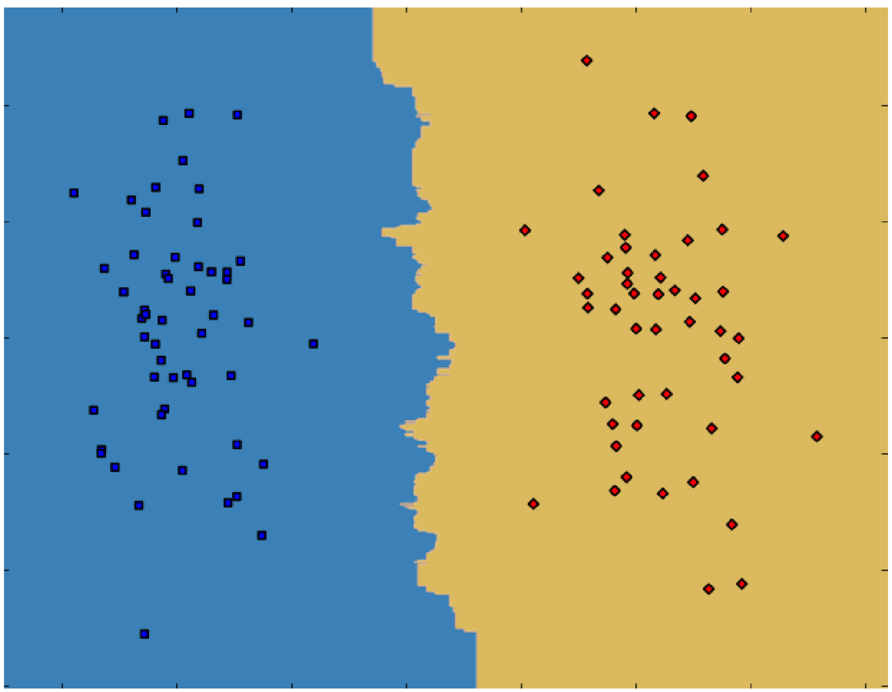


Various Data Decimations

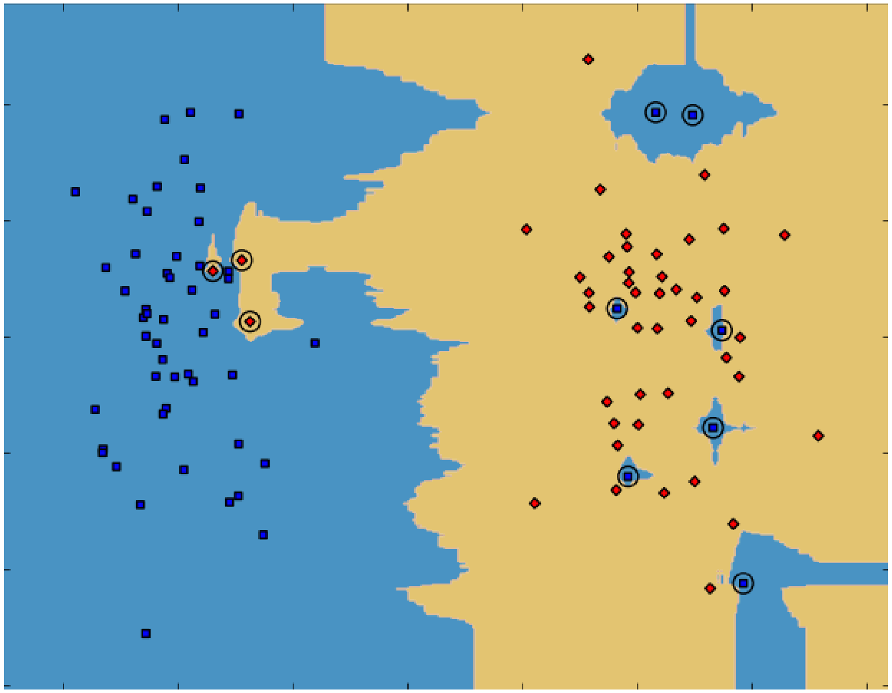
Outline

- "Adversarial" ambition is ambiguous (and alliterative).
- Machine learning has default expectations.
- These are deceptively subverted by **label tampering attacks**.
- There are a variety of possible label tampering attacks.
- "Quantified paranoia" might be one way to detect them.

Label Tampering; an "Algorithm-Aware" Attack



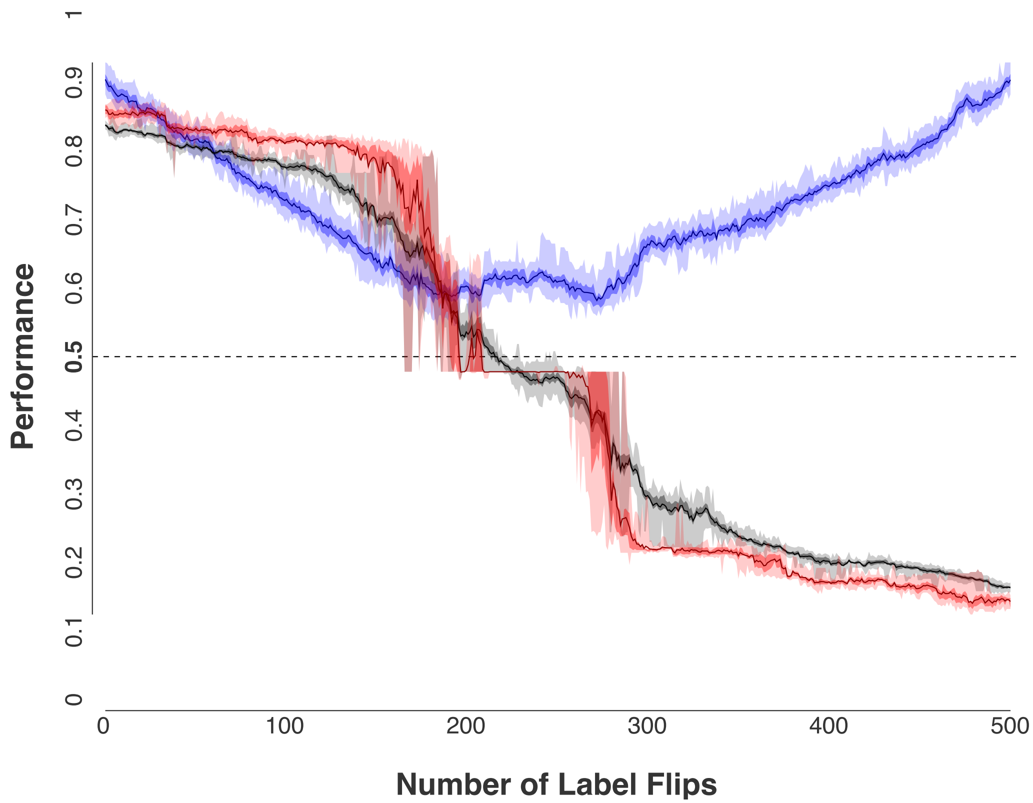
No Label Tampering



10% Random Label Tampering

Random Tampering

Mindless, random flipping of labels is eventually effective enough.



Pass/Fail Product Inspection

cross-validation on training, **ensemble performance on test**, **non-ensemble performance on test**.

How We Specify A Label Tampering Attack

- An "attack specification" is:
  - a specific sorting of all the training samples, in the order in which we'll tamper with the labels,
  - plus, for each sample, a specification of the tampered value we'll change it to.
- An "attack at budget N" is a set of training data where the truth labels of the first N samples in an attack have been altered according to a particular attack specification.
- An "attack heuristic" is a method for generating an attack specification from a set of training samples.

Original, Untampered Training Data

id	Truth	a_1	a_2	a_3	...	a_K
q_0	I	8	612	0.57	...	0.70
q_1	R	12	1003	0.97	...	0.12
q_2	R	99	2	0.33	...	0.03
q_3	I	3	27	0.12	...	0.13
q_4	R	16	183	0.08	...	0.58
q_5	I	17	665	0.36	...	0.64
q_6	I	44	1212	0.29	...	0.42
q_7	I	42	24	0.33	...	0.88
q_8	R	78	42	0.44	...	0.52
q_9	I	32	111	0.83	...	0.71

(Of course, no real training set would have just ten samples.)

An Attack Specification

id	Truth	Target	a_1	a_2	a_3	...	a_K
q_2	R	I	99	2	0.33	...	0.03
q_9	I	R	32	111	0.83	...	0.71
q_5	I	R	17	665	0.36	...	0.64
q_0	I	R	8	612	0.57	...	0.70
q_1	R	I	12	1003	0.97	...	0.12
q_6	I	R	44	1212	0.29	...	0.42
q_3	I	R	3	27	0.12	...	0.13
q_7	I	R	42	24	0.33	...	0.88
q_4	R	I	16	183	0.08	...	0.58
q_8	R	I	78	42	0.44	...	0.52

An Attack at Budget=4

id	Truth	Target	Tampered	a_1	a_2	a_3	$\ldots$	a_K
q_2	R	I	I	99	2	0.33	$\ldots$	0.03
q_9	I	R	R	32	111	0.83	$\ldots$	0.71
q_5	I	R	R	17	665	0.36	$\ldots$	0.64
q_0	I	R	R	8	612	0.57	$\ldots$	0.70
q_1	R	I	R	12	1003	0.97	$\ldots$	0.12
q_6	I	R	I	44	1212	0.29	$\ldots$	0.42
q_3	I	R	I	3	27	0.12	$\ldots$	0.13
q_7	I	R	I	42	24	0.33	$\ldots$	0.88
q_4	R	I	R	16	183	0.08	$\ldots$	0.58
q_8	R	I	R	78	42	0.44	$\ldots$	0.52

Now build an ML model with the “Tampered” column as the truth data.

Outline

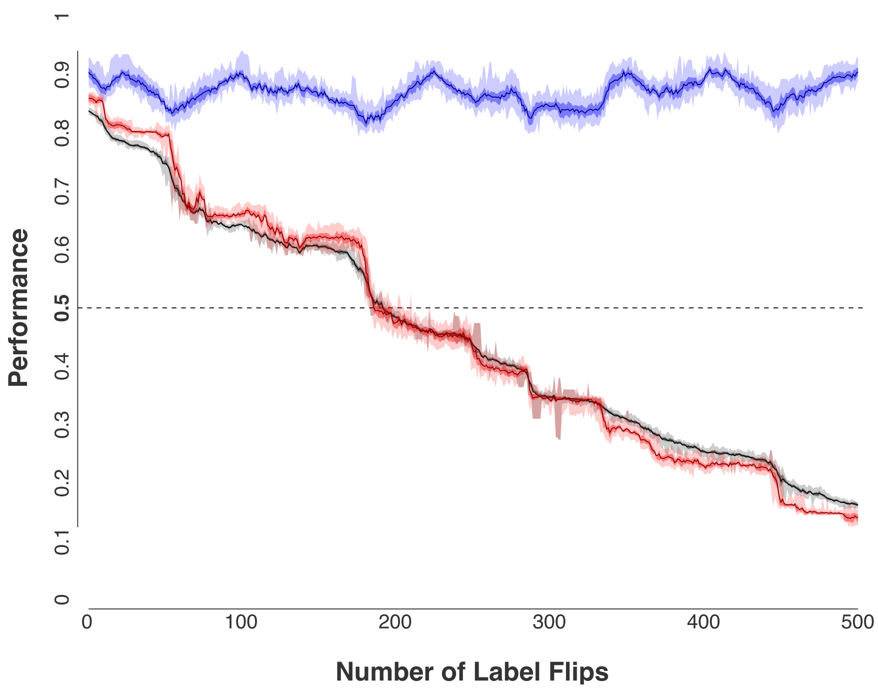
- "Adversarial" ambition is ambiguous (and alliterative).
- Machine learning has default expectations.
- These are deceptively subverted by label tampering attacks.
- There are a **variety of possible label tampering attacks**.
- "Quantified paranoia" might be one way to detect them.

The "Brute Clustering" Heuristic

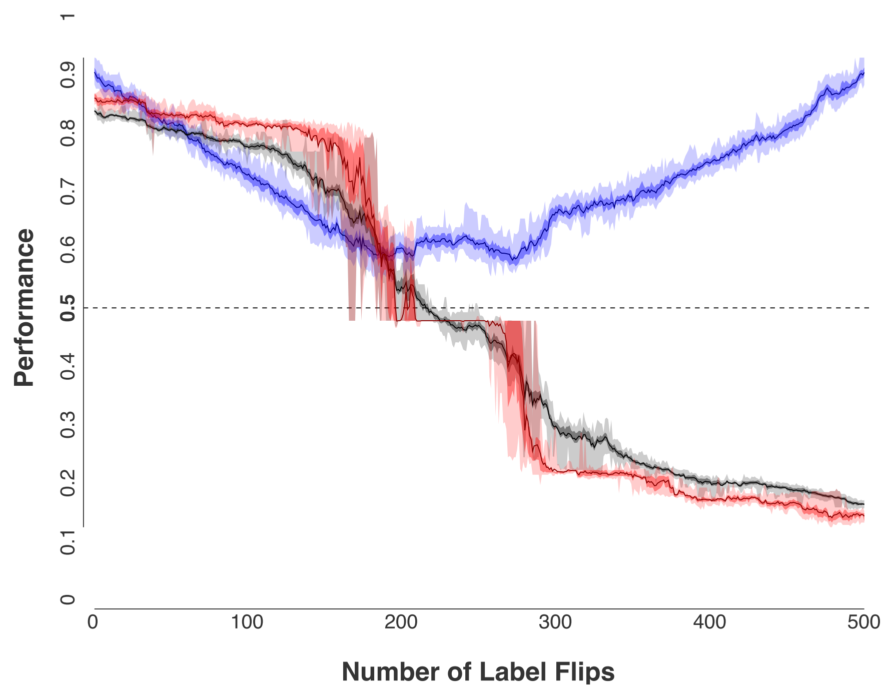
- The heuristic:
  - Do an unsupervised clustering of all training samples.
  - Pick an unattacked cluster at random.
  - Randomly order only the points in that cluster.
  - Repeat until all clusters have been attacked.

The "Brute Clustering" Heuristic

A smarter attack would try to suppress the cross-validation "dip" signature.



Brute Clustering



Random

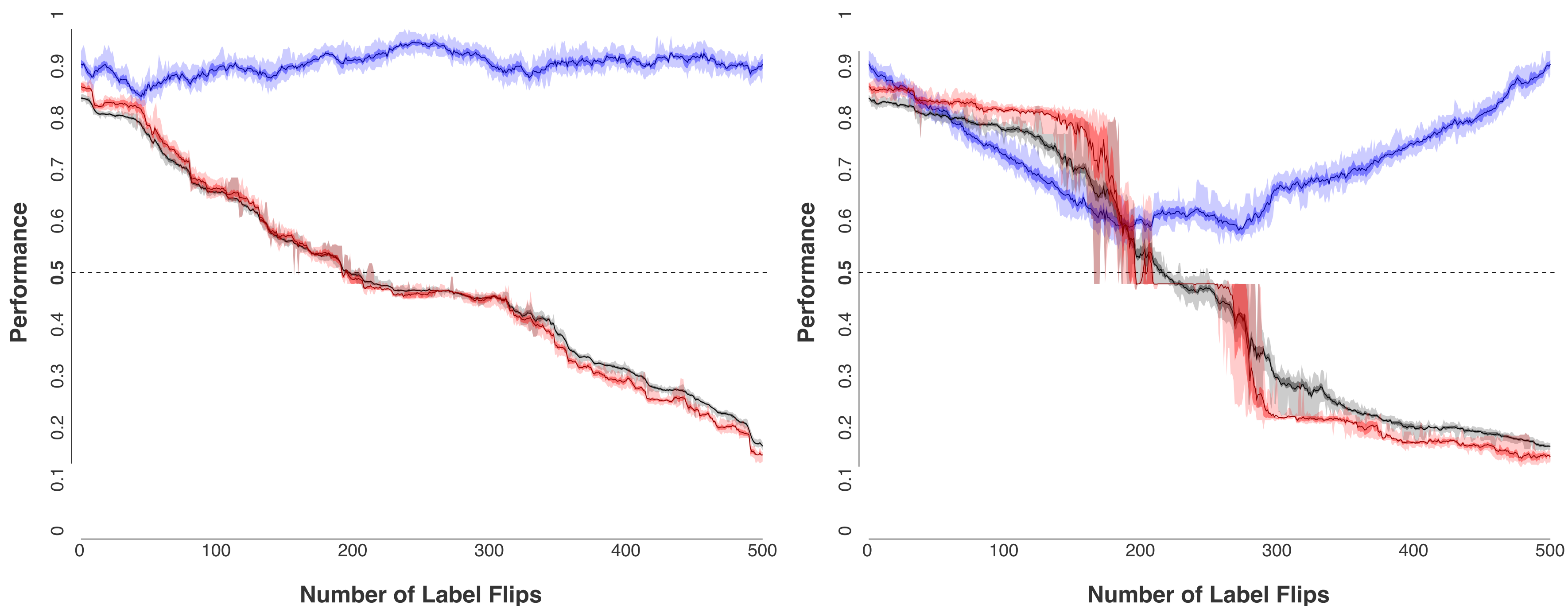
cross-validation on training, ensemble performance on test, non-ensemble performance on test.

The "Conditional Prediction Ordinate" Attack

- The heuristic:
  - Fit a logistic regression model, generate feature weights β_j.
  - Use β_j and Monte Carlo simulation to compute CPO_i[5] for each training sample i.
  - CPO_i is a measure of the sample i's influence on the model.
  - Sort by influence, attack most influential samples first.

Attack Statistically Significant Samples via CPO

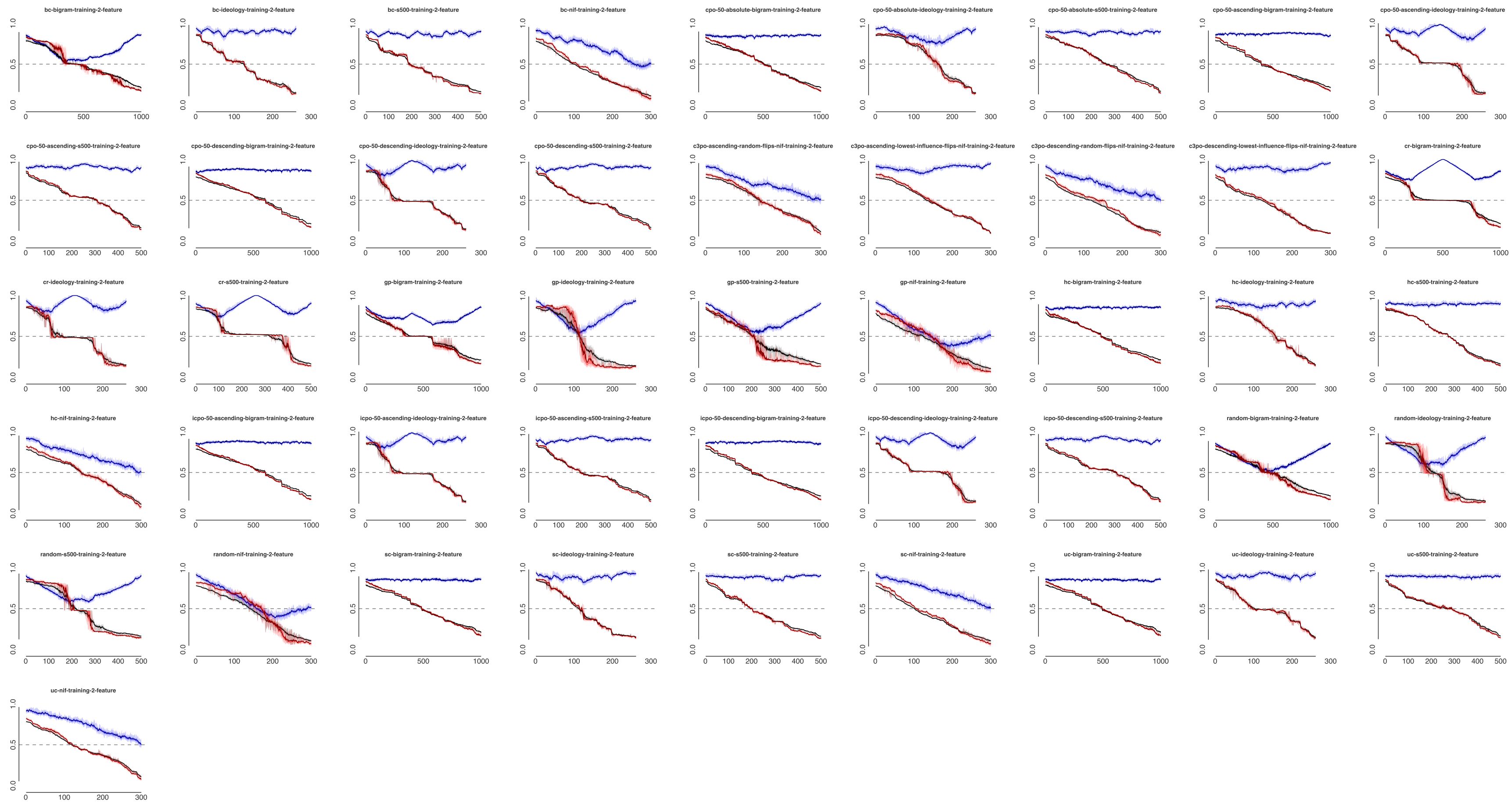
Even smarter attacks drive down accuracy, with less signaling.



Absolute CPO   Random

cross-validation on training, ensemble performance on test, non-ensemble performance on test.

We Invented Many Such Attacks



We Invented Many Such Attacks

Random: Attack samples in a random order.

Class Random: Pick a class randomly, change every sample of that class to some other random class. Repeat until all classes are attacked.

Greedy Pessimal: Iterative greedy search, attacking the training sample that reduces test performance the most at each iteration.

Brute Clustering: Cluster the samples, pick an unattacked cluster at random, attack its samples in a random order. Repeat.

Subtle Clustering: Cluster the samples, pick an unattacked cluster at random, attack its samples from the outside in. Repeat. (Outside in to promote stealth.)

Heterogeneous Clustering: Cluster the samples. Use a one-way chi-squared test to sort the clusters from most to least heterogeneous. Attack clusters in that order, in each case attacking from the inside out. (Inside out to sow confusion as quickly as possible.)

Understated Clustering: Cluster the samples. Sort the clusters in descending order by their percentage population of a target class L_c. Attack clusters in that order, in each case attacking from the outside in. (Sort by class to specifically try to confuse detection of a target class.)

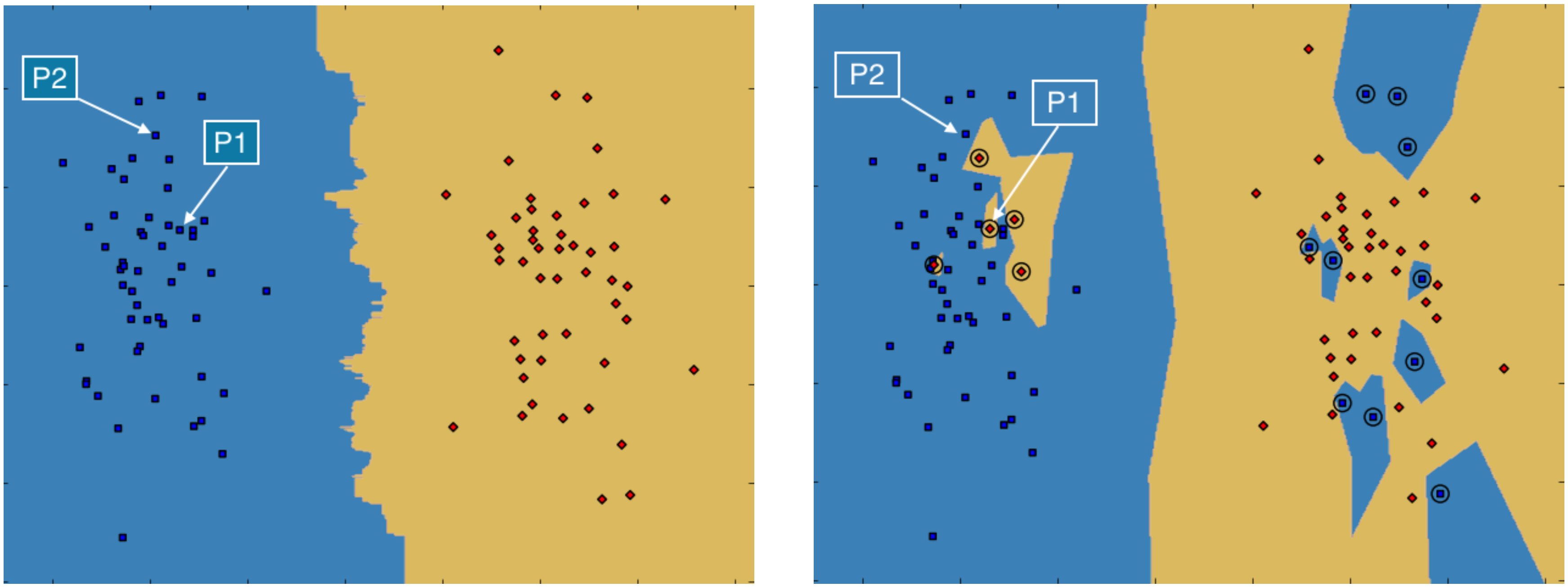
Conditional Predictive Ordinates: Compute the conditional predictive ordinate (CPO) of every sample, as that's an inverse measure of the influence of that sample. Attack in order of increasing absolute value of CPO.

Outline

- "Adversarial" ambition is ambiguous (and alliterative).
- Machine learning has default expectations.
- These are deceptively subverted by label tampering attacks.
- There are a variety of possible label tampering attacks.
- **"Quantified paranoia"** might be one way to detect them.

Not Discussing EOM Remediation Today

EOM: Ensembles of Outlier Measures



No Label Tampering — 15% Label Tampering

One example outlier measure: K nearest neighbor agreement (nna)

- p1 (pre-attack): 5-nna is 1.0
- p2 (pre-attack): 5-nna is 1.0

- p1 (tampered): 5-nna is 0.2
- p2 (untampered): 5-nna is 0.8

Quantified Paranoia Via Pseudo-Bayes Factors

- Fit a model M_a on untampered data A.
- Fit a model M_b on possibly tampered data B.
- Don't ask: "are the models similar?"
- Do ask: "are the model *fits on* B similar".

Pseudo-Bayes Factors[8]

- CPO_i are goodness-of-fit measures; they track *outliers*.

- Intuition: If B is untampered data drawn from the same distribution as A, then Models A and B should both individually have roughly the same goodness-of-fit for B.

- PBF is the ratio of those model fits:
  - If the fits of Models A and B on data B are indeed nearly identical, the PBF will be very close to 1.
  - If B has been tampered with, if it is different than A, then Model B will fit B better than Model A, Model B will have fewer outliers, and the ratio will be higher than 1.

An Example Empirical Experiment

Budget	Random	CPO	SC
0	0.19		
1	3.55	4.17	2.17
2	5.11	8.57	4.77
3	6.96	12.21	6.07
4	11.44	15.68	5.99
5	15.62	18.03	8.06
6	17.43	19.80	10.13
7	20.67	20.77	11.72
8	23.00	22.60	13.70
9	24.64	24.12	13.64
10	24.26	26.82	13.33
11	24.93	28.10	14.96
12	26.65	29.70	16.57

log(PBF) comparison of three attacks

Date	10/1	10/7	11/1	11/7	12/1	12/7
log(PBF)	0.00	0.23	-0.05	0.11	0.55	-0.11

PBF over time with unattacked, naturally evolving data

Interpretation	log(PBF)
Very strong support for tampering in A	<-5
Strong support for tampering in A	-5 to -3
Positive support for tampering in A	-3 to -1
Weak support for tampering in A	-1 to 0
No support for tampering in A	0
Weak support for tampering in B	0 to 1
Positive support for tampering in B	1 to 3
Strong support for tampering in B	3 to 5
Very strong support for tampering in B	>5

Interpretation of log(PBF)[8]

- Untampered model from set-aside data.
- Budget "0" is the "untampered data" case.
- Caveat: only 260 data points, so three tampered data points is 1%.

Final Summary

- "Adversarial" ambition is ambiguous (and alliterative).
- Machine learning has default expectations.
- These are deceptively subverted by label tampering attacks.
- There are a variety of possible label tampering attacks.
- "Quantified paranoia" might be one way to detect them.

End Notes

Collaborators: Sandians: Ali Pinar, Dave Zage, Jon Crussell, Katie Rodhouse, Dave Robinson, Warren Davis, Justin (JD) Doak, Jeremy Wendt, Curtis Johnson. Others: Rich Colbaugh, Kristin Glass, Brian Jones, Yevgeniy Vorobeychik, Jeff Shelburg.

References

[1] Banfield, R. E., Hall, L. O., Bowyer, K. W., Bhadoria, D., Kegelmeyer, W. P., and Eschrich, S. A comparison of ensemble creation techniques. In *Proceedings of the Fifth International Conference on Multiple Classifier Systems, MCS2004* (2004), J. K. F. Roli and T. Windeatt, Eds., vol. 3077 of *Lecture Notes in Computer Science*, Springer-Verlag.

[2] Chawla, N. V., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. Learning ensembles from bites: A scalable and accurate approach. *Journal of Machine Learning Research 5* (2004), 421–451.

[3] Chesterton, G. *The Man Who Was Thursday: A Nightmare*. Jovian Press, 1908.

[4] Gaudrillot-Roy, Z. Dissections. Artist's Site. `http://www.designboom.com/art/architectural-dissections-zacharie-gaudrillot-roy-isolate-building-facades-03-10-2013/`.

[5] Gelfand, A. E., Dey, D. K., and Chang, H. Model determination using predictive distributions with implementation via sampling-based methods (with discussion). *Bayesian Statistics 4* (1992), 147–167.

[6] Goodfellow, I. J. NIPS 2016 tutorial: Generative adversarial networks. *CoRR abs/1701.00160* (2017).

[7] Kegelmeyer, P., Shead, T. M., Crussell, J., Rodhouse, K., Robin-son, D., Johnson, C., Zage, D., Davis, W., Wendt, J., Doak, J. J., Cayton, T., Colbaugh, R., Glass, K., Jones, B., and Shelburg, J. Counter adversarial data analytics. Tech. rep., Sandia National Laboratories, 2015.

[8] Lodewyckx, T., Kim, W., Lee, M. D., Tuerlinckx, F., Kuppens, P., and Wagenmakers, E.-J. A tutorial on Bayes factor estimation with the product space method. *Journal of Mathematical Psychology 55*, 5 (Oct. 2011), 331–347.

[9] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (New York, NY, USA, 2017), ASIA CCS '17, ACM, pp. 506–519.

[10] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction apis. *CoRR abs/1609.02943* (2016).

Supplemental Slides

Conditional Prediction Ordinate Math

- Logistic regression:
  - Assume $P(y = 1|\beta, \mathbf{x}_i) = \psi\left(\sum_j \beta_j x_{i,j}\right)$.
  - Use a logistic function for ψ: $\psi(z) = \frac{\exp(z)}{1+\exp(z)}$.
- Conditional Prediction Ordinate
  - CPO_i is the inverse of the posterior mean of the inverse likelihood of y_i:

$$CPO_i = \frac{f(y)}{f(y_{\neg i})} = \left(\frac{1}{N}\sum_{j=1}^{N}\frac{1}{f(y_i|\beta_j)}\right)^{-1}$$

  - CPO_i is posterior probability of y_i when the model is fitted to all data *except* y_i.
  - If $|CPO_i|$ is high, y_i is not surprising, is as expected.
  - If $|CPO_i|$ is low, y_i *is* surprising, is an influential sample, is not well modeled by $f(y_{\neg i})$, and so would have changed the $f(y)$ model if present.

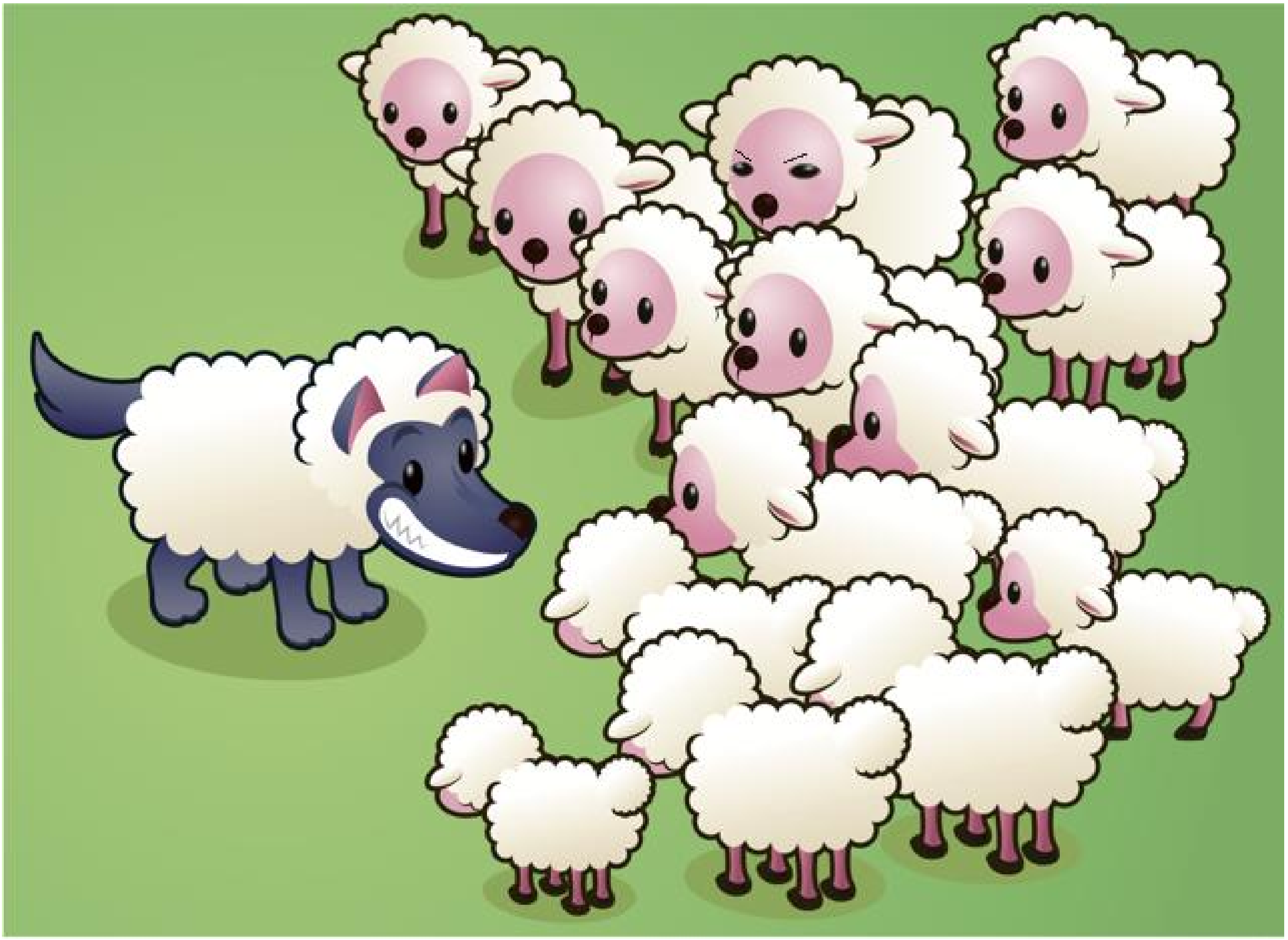
Pseudo-Bayes Factors[8]

- CPO_i are goodness-of-fit measures; they track outliers.
- Intuition: If B is untampered data drawn from the same distribution as A, then Models A and B should both individually have roughly the same goodness-of-fit for B. We can check this by examining the CPO values generated by Models A and B on B.

$$PBF_{ab} = \frac{f(B|M_a)}{f(B|M_b)} = \frac{\int f(B|\beta_a, M_a) f(\beta_a|M_a) d\beta_a}{\int f(B|\beta_b, M_b) f(\beta_b|M_b) d\beta_b} = \frac{\prod_N CPO_{ai}|M_a}{\prod_N CPO_{bi}|M_b}$$

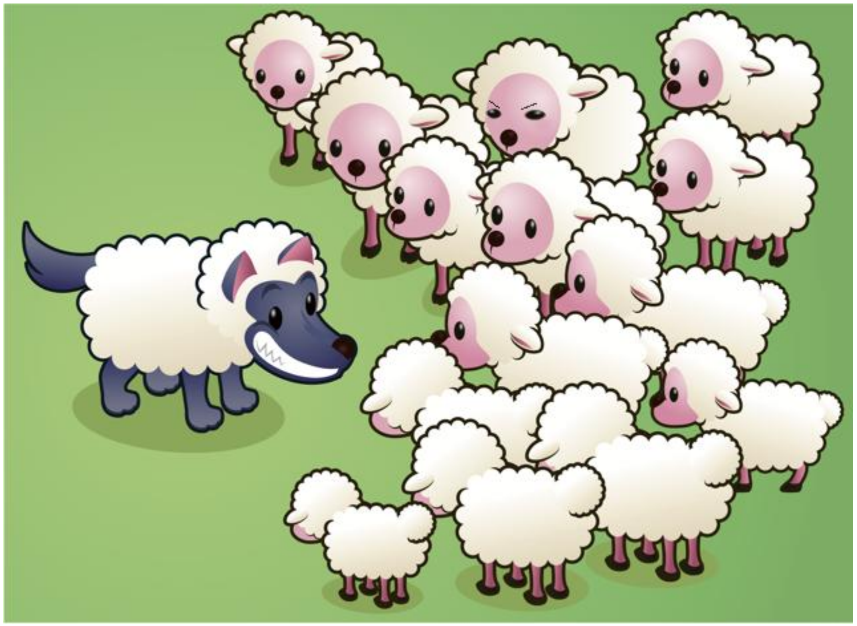
- If the fits of Models A and B on data B are indeed nearly identical, the PBF will be very close to 1.
- If B has been tampered with, if it is different than A, then Model B will fit B better than Model A, Model B will have fewer outliers, and the ratio will be higher than 1.

Counter Adversarial Data Analytics (CADA)



IF (white AND fuzzy) Then <Harmless>

Counter Adversarial Data Analytics (CADA)



IF (white AND fuzzy) Then <Harmless>

Sandia makes **critical use of data analytics**, which our adversaries therefore **seek to sap, even suborn.**
Through **understanding our methods,** they seek to produce data which is evolving, incomplete, deceptive, and otherwise **custom-designed to defeat our analysis.**
We **cannot prevent this:** we frequently must depend on data over which our adversaries have extensive influence.
We will thus develop and assess novel data analysis methods to **counter that adversarial influence.**

- Goals:
  - Discover generalizable, quantifiable counter-adversarial principles.
  - Specifically: investigate a) robust, b) predictive, and c) dynamic defenses.
  - Convert them to relevant, realistic methods with practical implementations.

Philosophy

"We must learn to love life without ever trusting it." (G.K. Chesterton)

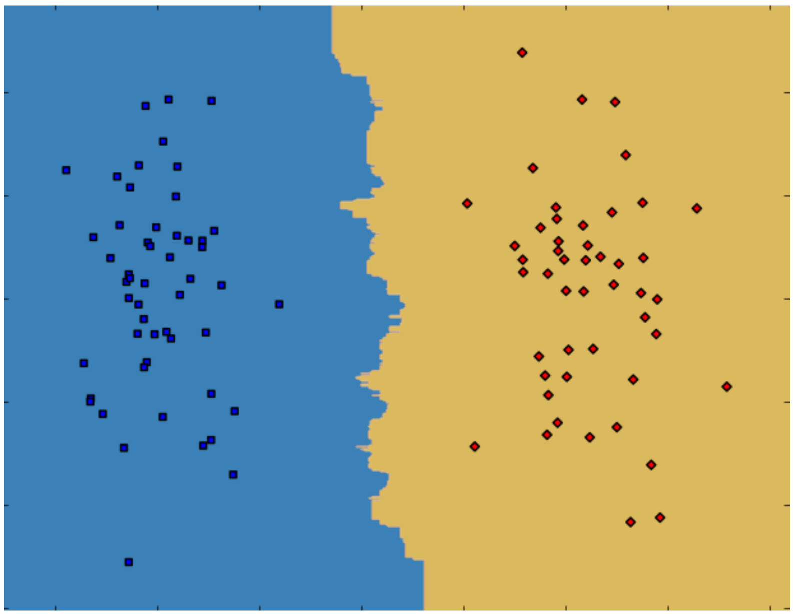
⇒ "We must learn to love ~~life~~ data without ever trusting it."

CADA is working to turn this into *quantified*, practical advice.

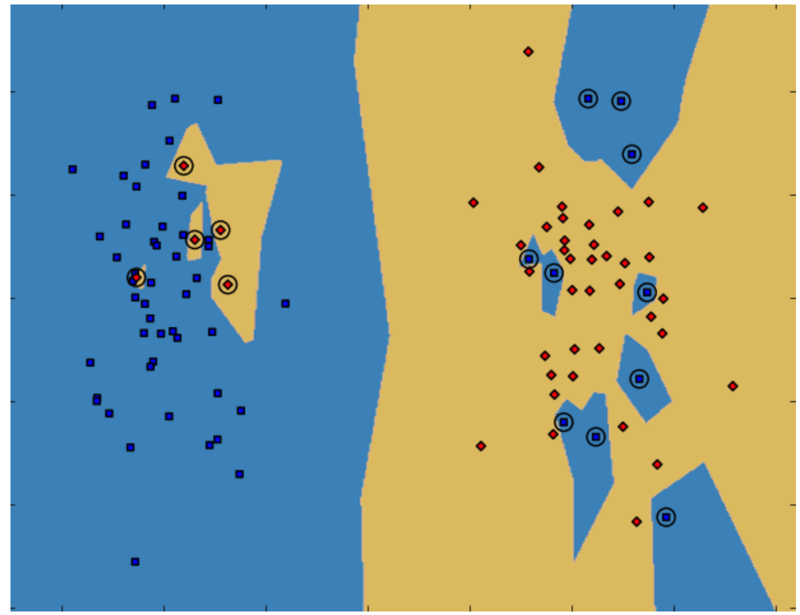
- Data Sciences Research Challenge late start LDRD; started April 2013
- 1.25M over 1.5 years, with staff across two sites and five divisions.
- Coordinates with Sandia LDRDs (HostWatch, Alert Triage, MaLAdE) and program work (Mountain Creek).
- Nascent external work on the effects of data tampering[**?**, **?**].

Is There Any Way to Mitigate the Damage?

EOM: Ensembles of Outlier Measures



No Label Tampering



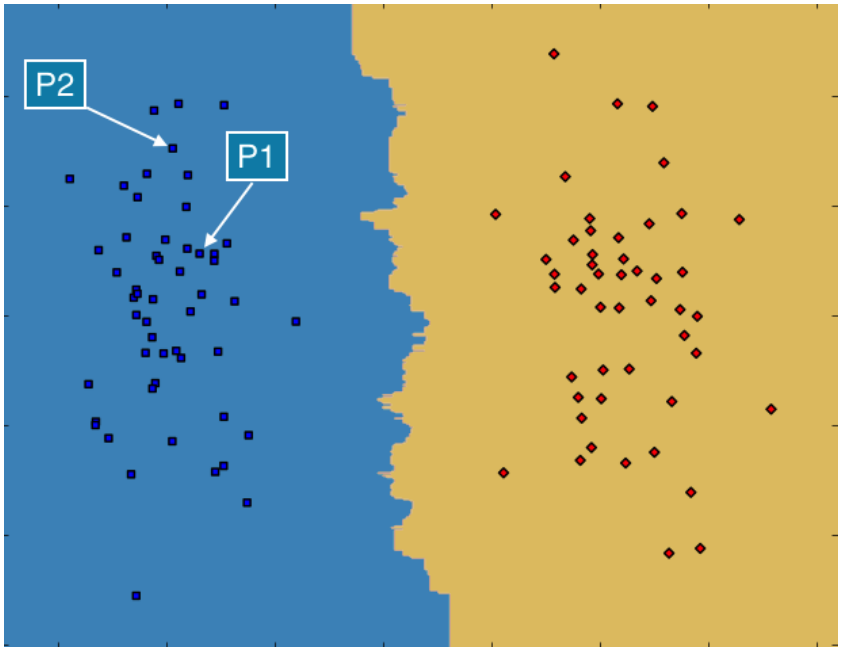
15% Label Tampering

(Circles indicate label-tampered points.)

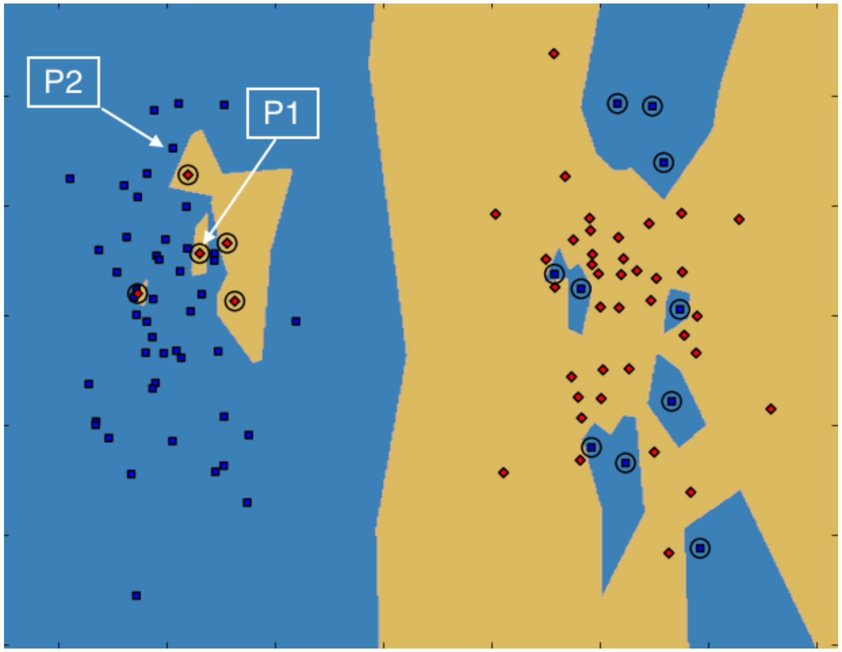
Outliers (weakly) signal tampering.
But no one outlier measure is perfect. So . . .
use a *variety* of outlier measures, at a variety of parameter settings,
and interpret them with ensembles of decision trees.

Is There Any Way to Mitigate the Damage?

EOM: Ensembles of Outlier Measures



No Label Tampering



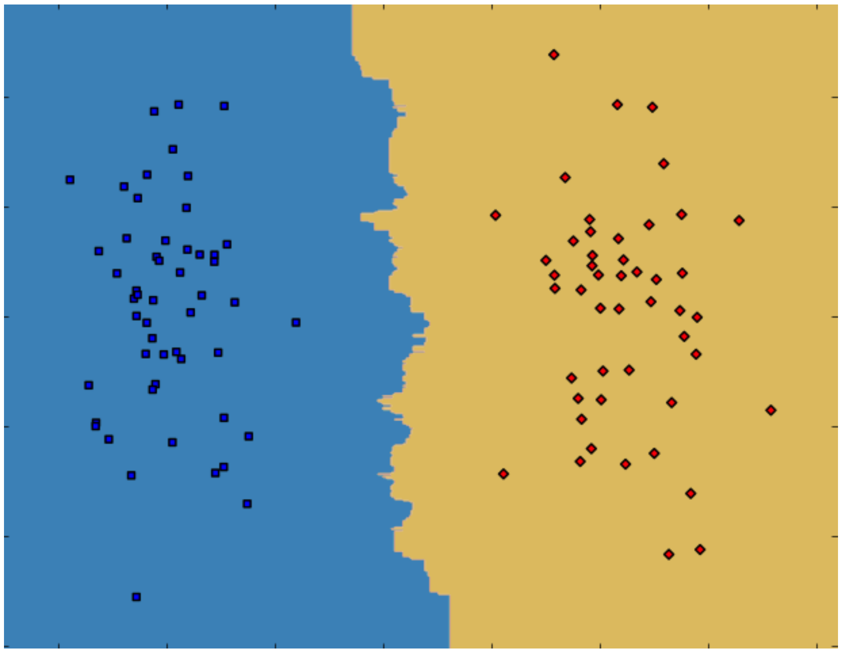
15% Label Tampering

One example outlier measure: K nearest neighbor agreement (nna)

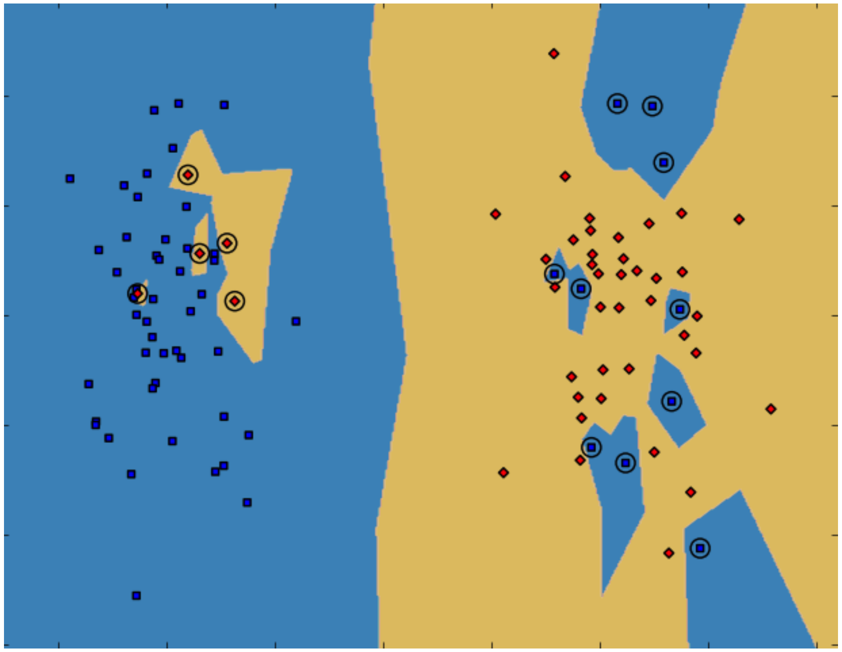
- p1 (pre-attack): 5-nna is 1.0
- p2 (pre-attack): 5-nna is 1.0

- p1 (tampered): 5-nna is 0.2
- p2 (untampered): 5-nna is 0.8

Current Outlier Measures



No Label Tampering



15% Label Tampering

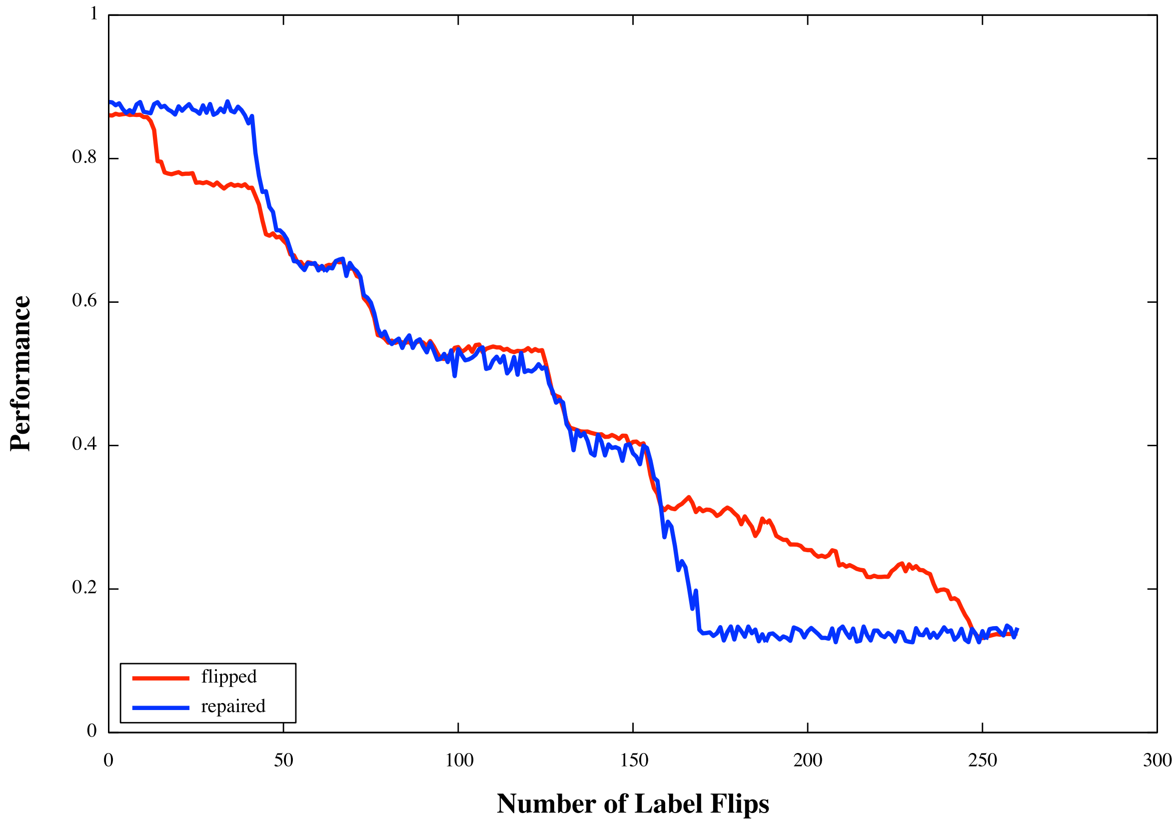
- Label Spreading
- KNN agreement
- Local Outlier Factor

- Boosting weights
- Confidence mismatch
- Local Correlation Integral (LOCI)
- DBSCAN

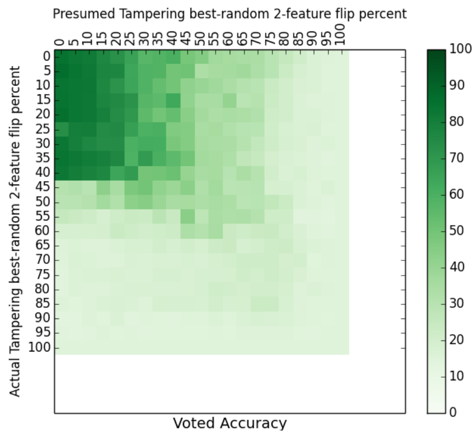
Detect and Repair Tampering

"Flipped": The tampered data.

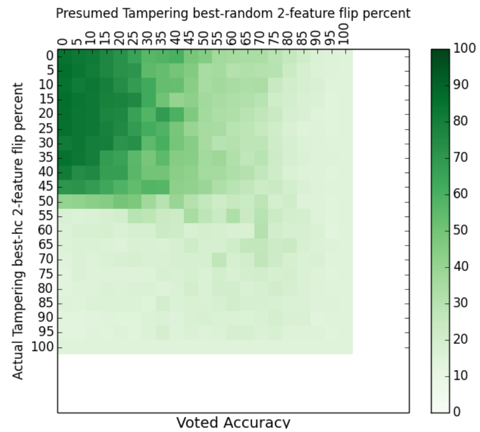
"Repair": wherever tampered labels are detected, *correct the label*.



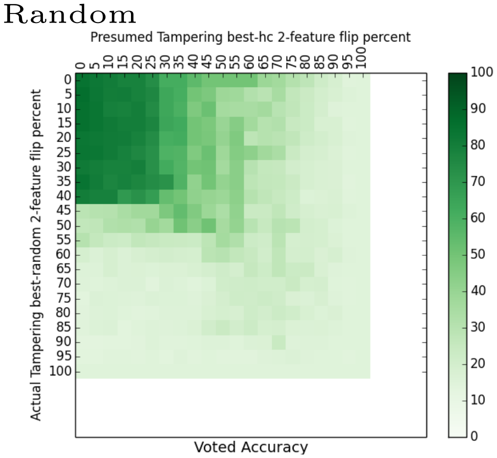
A Mosaic of Tamper Remediations



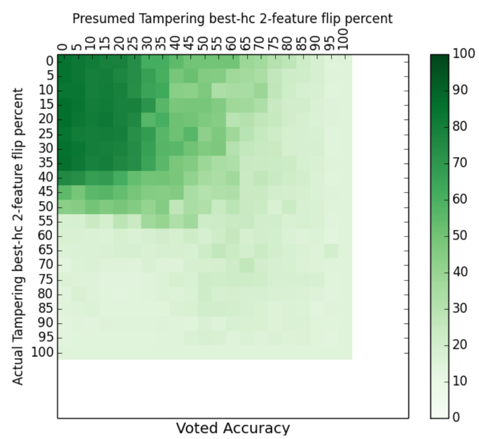
Assume Random, Actually Random



Assume Random, Actually HC



Assume HC, Actually Random



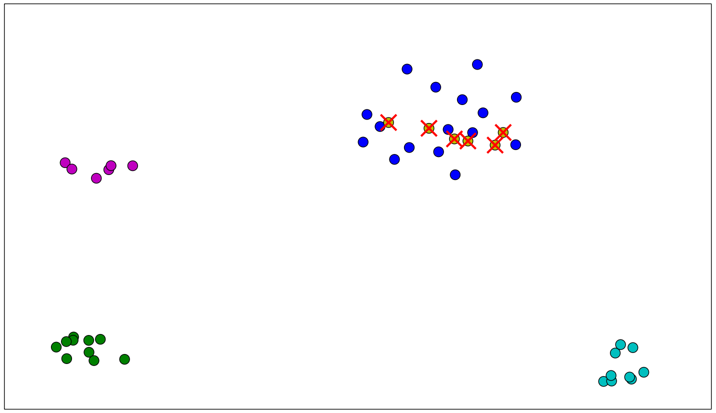
Assume HC, Actually HC

Summary

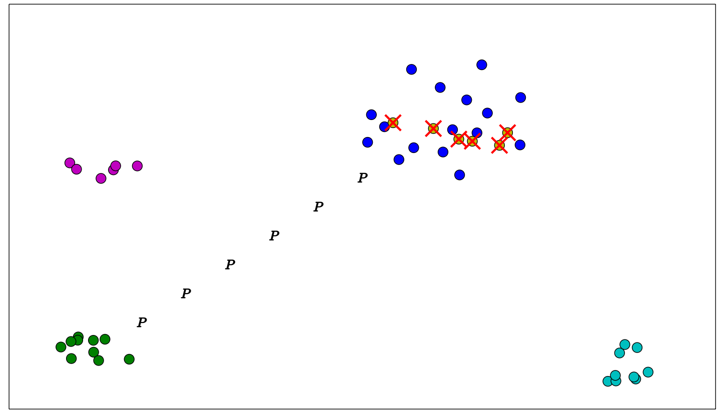
Summary

EOM Applied to a Very Different Analytic

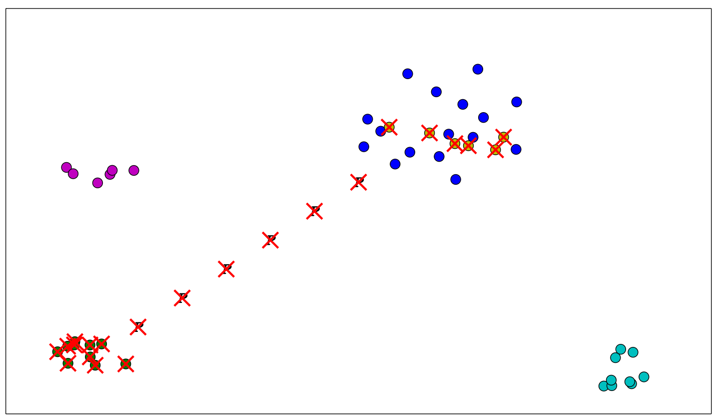
Clustering as applied to Android app plagiarism detection[**?**].



Red X: indicates plagiarized apps



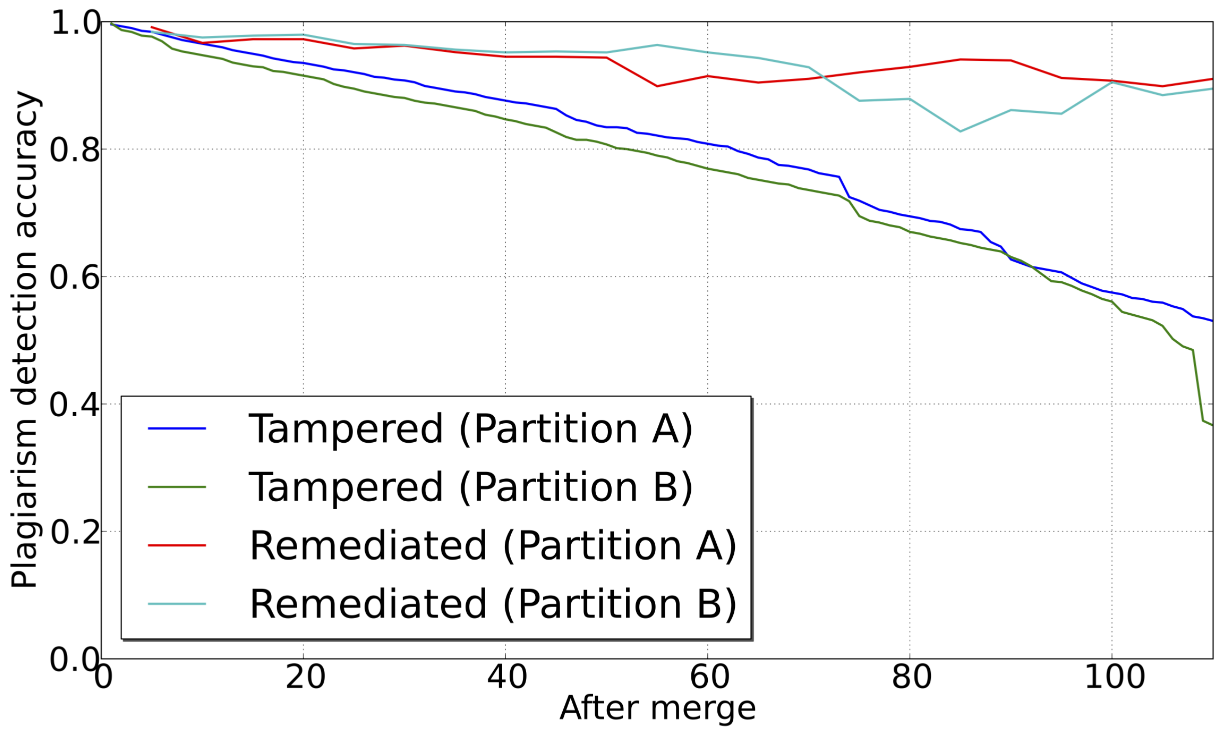
Black P: fake apps



Red X: true plagiarisms *and* false alarms

EOM is Gratifyingly General

- Applied to clustering, not supervised machine learning, with …
- … an *entirely* different set of outlier features.
- Yet still: poisoned data can be found and removed via EOM.



Summary

Machine learning has default expectations.

These are subverted, even deceptive in the face of label tampering attacks.

"Ensembles of Outlier Methods" can help detect and mitigate those attacks,

And in a surprisingly general way.

(Plus, a **fledgling** example of a graph analysis attack.)