Responsible Machine Learning*

Lecture 4: Security

Patrick Hall

The George Washington University

June 3, 2020

*This material is shared under a CC By 4.0 license which allows for editing and redistribution, even for commercial purposes. However, any derivative work should attribute the author.

Contents

Overview

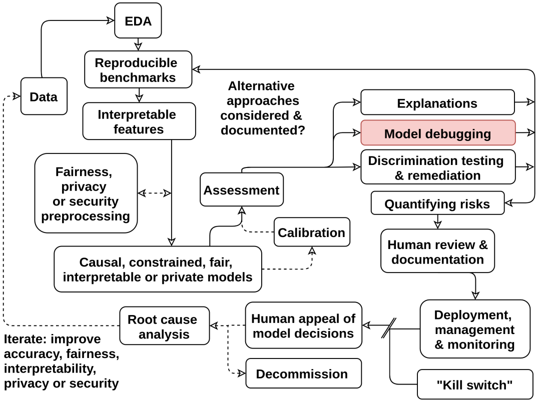
Attacks

General Concerns

General Solutions

Summary

A Responsible Machine Learning Workflow†



† *A Responsible Machine Learning Workflow*

Why Attack Machine Learning Models?

Hackers, malicious or extorted insiders, and their criminal associates or organized extortionists, seek to:

- induce beneficial outcomes from a predictive or pattern recognition model or induce negative outcomes for others.
- commit corporate espionage.
- steal intellectual property including models and data.

Types of Security Risks and Attacks

- Data Poisoning
- Backdoors and Watermarks
- Surrogate Model Inversion
- Membership Inference
- Adversarial Example
- Impersonation Attacks

Data Poisoning Attacks: **What?**

- Hackers gain unauthorized access to training data and alter it before model training or retraining.
- Malicious or extorted data science or IT insiders do the same while working at a ...
  - small disorganized firm where the same person is allowed to manipulate training data, train models, and deploy models.
  - massive firm, and covertly accumulate the permissions needed to manipulate training data, train models, and deploy models.

Data Poisoning Attacks: **How?**

Attributes of attacker

dti: 10.4
fico: 690
m_delinq: 4
⋮

dti	fico	m_delinq		deny
0.9	740	0	⋮	0
9	680	4	⋮	1
7.2	700	3	⋮	1
2.3	790	0	⋮	0

Original training data

attack

dti	fico	m_delinq		deny
0.9	740	0	⋮	0
9	680	4	⋮	0
7.2	700	3	⋮	1
2.3	790	0	⋮	0

Altered training data

Attacker alters data before model training to ensure favorable outcomes.

Data Poisoning Attacks: **Defenses**

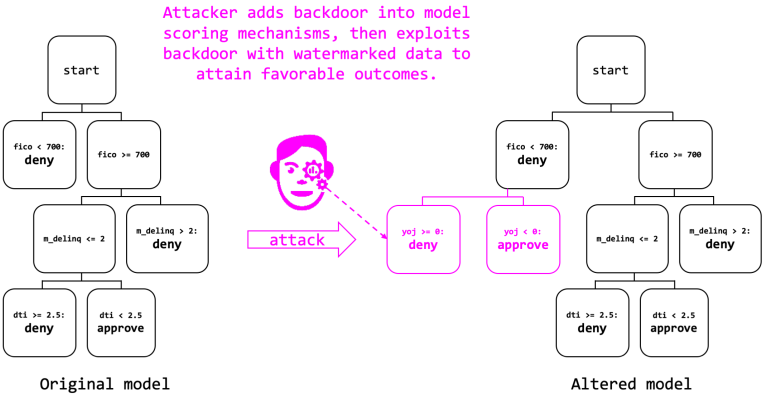
- **Disparate impact analysis**: Use tools like aequitas, AIF360, or your own fair lending tools, to look for discrimination in your model's predictions.
- **Fair or private models**: E.g. learning fair representations (LFR), private aggregation of teacher ensembles (PATE) [5], [9].
- **Reject on negative impact (RONI) analysis**: See: *The Security of Machine Learning* [2].
- **Residual analysis**: especially for large positive deviance residuals.
- **Self-reflection**: Score your models on your employees, consultants, and contractors and look for anomalously beneficial predictions.

Backdoors and Watermarks: **What?**

- Hackers gain unauthorized access to your production scoring code OR ...
- Malicious or extorted data science or IT insiders change your production scoring code ...

... adding a backdoor that can be exploited using water-marked data.

Backdoors and Watermarks: **How?**



Backdoors and Watermarks: **Defenses**

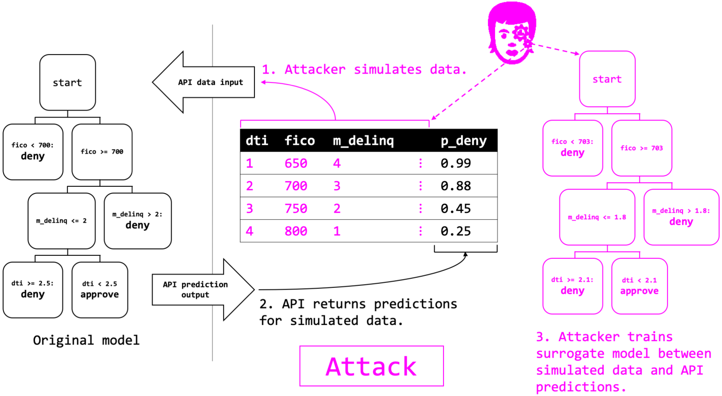
- **Anomaly detection**: Screen your production scoring queue with an autoencoder, a type of machine learning (ML) model that can detect anomalous data.
- **Data integrity constraints**: Don't allow impossible or unrealistic combinations of data into your production scoring queue.
- **Disparate impact analysis**: See Slide 8.
- **Version control**: Track your production model scoring code just like any other enterprise software.

Surrogate Model Inversion Attacks: **What?**

Due to lax security or a distributed attack on your model API or other model endpoint, hackers or competitors simulate data, submit it, receive predictions, and train a surrogate model between their simulated data and your model predictions. This surrogate can ...

- expose your proprietary business logic, i.e. "model stealing" [8].
- reveal sensitive aspects of your training data.
- be the first stage of a membership inference attack (see Slide 16).
- be a test-bed for adversarial example attacks (see Slide 19).

Surrogate Model Inversion Attacks: **How?**



dti	fico	m_delinq	⋮	p_deny
1	650	4	⋮	0.99
2	700	3	⋮	0.88
3	750	2	⋮	0.45
4	800	1	⋮	0.25

Surrogate Model Inversion Attacks: **Defenses**

- **Authentication**: Authenticate users of your model's API or other endpoints.
- **Defensive watermarks**: Add subtle or unusual information to your model's predictions to aid in forensic analysis if your model is hacked or stolen.
- **Throttling**: Consider artificially slowing down your prediction response times, especially after anomalous behavior is detected.
- **White-hat surrogate models**: Train your own surrogate models as a white-hat hacking exercise to see what an attacker could learn about your public models.

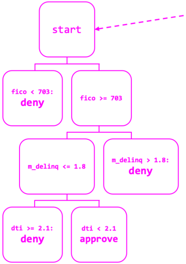
Membership Inference Attacks: **What?**

Due to lax security or a distributed attack on your model API or other model endpoint ...

- this two-stage attack begins with a surrogate model inversion attack (see Slide: 13).
- A second-level surrogate is then trained to discriminate between rows of data in, and not in, the first-level surrogate's training data.
- The second-level surrogate can dependably reveal whether a row of data was in, or not in, your original training data [7].

Simply knowing if a person was in, or not in, a training dataset can be a violation of individual or group privacy. However, when executed to the fullest extent, a membership inference attack can allow a bad actor to **rebuild your sensitive training data**!

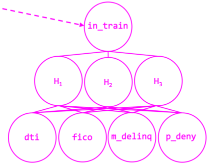
Membership Inference Attacks: **How?**



1. Attack begins with trained surrogate model.

dti	fico	m_delinq		p_deny	in_train
1	650	4	⋮	0.96	1
2	700	3	⋮	0.89	1
3	750	2	⋮	0.44	1
4	800	1	⋮	0.27	1
4	850	1	⋮	0.33	0
3	800	2	⋮	0.66	0
2	750	3	⋮	0.66	0
1	700	4	⋮	0.66	0

2. Attacker augments surrogate training data with new rows not in the original surrogate training data and scores them with the surrogate model.



3. Attacker trains a second-level surrogate model that can discriminate between rows in your training data and not in your training data.

Membership Inference Attacks: **Defenses**

- See Slide 14.
- **Monitor for training data**: Monitor your production scoring queue for data that closely resembles any individual used to train your model. Real-time scoring of rows that are extremely similar or identical to data used in training, validation, or testing should be recorded and investigated.

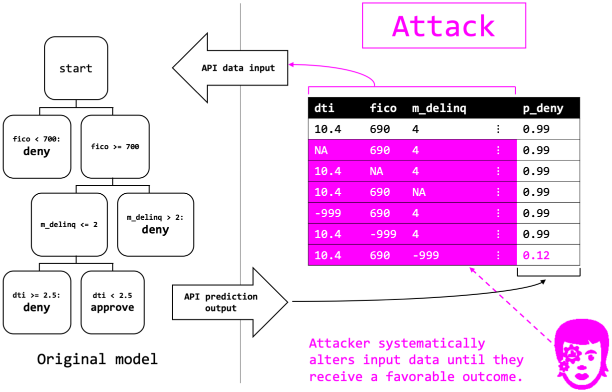
Adversarial Example Attacks: **What?**

Due to lax security or a distributed attack on your model API or other model endpoint, hackers or competitors simulate data, submit it, receive predictions, and learn by systematic trial-and-error ...

- your proprietary business logic.
- how to game your model to dependably receive a desired outcome.

Adversarial example attacks can also be enhanced, tested, and hardened using models trained from surrogate model inversion attacks (see Slide 13).

Adversarial Example Attacks: **How?**



Adversarial Example Attacks: **Defenses**

- **Anomaly detection**: See Slide 11.
- **Authentication**: See Slide 14.
- **Benchmark models**: Always compare complex model predictions to trusted linear model predictions. If the two model's predictions diverge beyond some acceptable threshold, review the prediction before you issue it.
- **Fair or private models**: See Slide 8.
- **Throttling**: See Slide 14.
- **Model monitoring**: Watch your model in real-time for strange prediction behavior.
- **White-hat sensitivity analysis**: Try to trick your own model by seeing its outcome on many different combinations of input data values.
- **White-hat surrogate models**: See Slide 14.

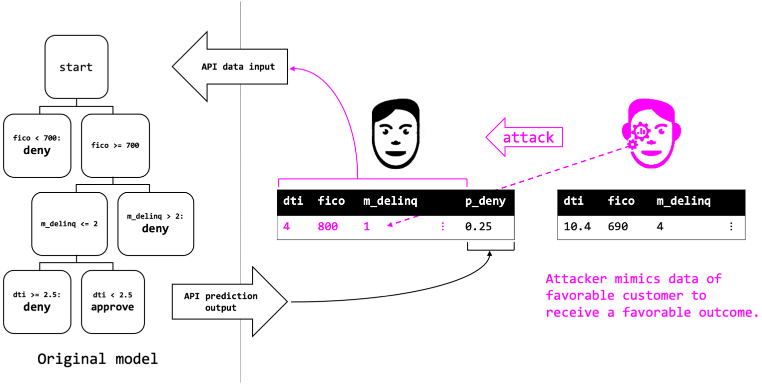
Impersonation Attacks: **What?**

Bad actors learn ...

- by inversion or adversarial example attacks (see Slides 13, 19), the attributes favored by your model and then impersonate them.
- by disparate impact analysis (see Slide 8), that your model is discriminatory (e.g. Propublica and COMPAS, Gendershades and Rekognition), and impersonate your model's privileged class to receive a favorable outcome.‡

‡This presentation makes no claim on the quality of the analysis in Angwin et al. (2016), which has been criticized, but is simply stating that such cracking is possible [1], [3].

Impersonation Attacks: **How?**



Impersonation Attacks: **Defenses**

- **Authentication**: See Slide 14.
- **Disparate impact analysis**: See Slide 8.
- **Model monitoring**: Watch for too many similar predictions in real-time. Watch for too many similar input rows in real-time.

General concerns

- **Black-box models**: Over time a motivated, malicious actor could learn more about your own black-box model than you know and use this knowledge imbalance to attack your model [4].
- **Black-hat eXplainable AI (XAI)**: While XAI can enable human learning from machine learning, regulatory compliance, and appeal of automated decisions, it can also make ML hacks easier and more damaging [6].
- **Distributed-denial-of-service (DDOS) attacks**: Like any other public-facing service, your model could be attacked with a DDOS attack.
- **Distributed systems and models**: Data and code spread over many machines provides a larger, more complex attack surface for a malicious actor.
- **Package dependencies**: Any package your modeling pipeline is dependent on could potentially be hacked to conceal an attack payload.

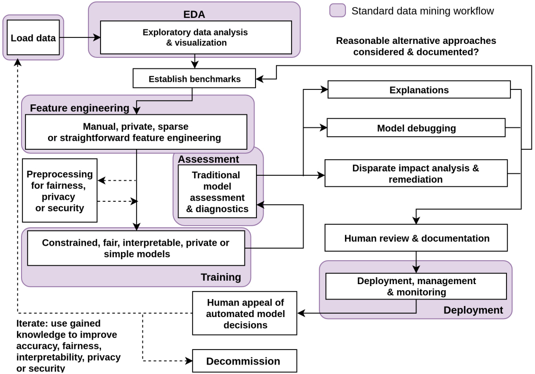
General Solutions

- **Authenticated access and prediction throttling**: for prediction APIs and other model endpoints.
- **Benchmark models**: Compare complex model predictions to less complex (and hopefully less hackable) model predictions. For traditional, low signal-to-noise data mining problems, predictions should not be too different. If they are, investigate them.
- **Encrypted, differentially private, or federated training data**: Properly implemented, these technologies can thwart many types of attacks. Improperly implemented, they simply create a broader attack surface or hinder forensic efforts.
- **Interpretable, fair, or private models**: In addition to models like LFR and PATE, also checkout monotonic GBMs, Rulefit, AIF360, and the Rudin group at Duke.

General Solutions

- **Model documentation, management, and monitoring**:
  - Take an inventory of your predictive models.
  - Document production models well-enough that a new employee can diagnose whether their current behavior is notably different from their intended behavior.
  - Know who trained what model, on what data, and when.
  - Monitor and investigate the inputs and predictions of deployed models on live data.
- **Model debugging and testing, and white-hat hacking**: Test your models for accuracy, fairness, and privacy before deploying them. Train white-hat surrogate models and apply XAI techniques to them to see what hackers can see.
- **System monitoring and profiling**: Use a meta anomaly detection system on your entire production modeling system's operating statistics — e.g. number of predictions in some time period, latency, CPU, memory and disk loads, number of concurrent users, etc. — then closely monitor for anomalies.

A Blueprint for Low-Risk Machine Learning [§]



[§] See: https://github.com/jphall663/hc_ml for more information.

Summary

- ML hacking is still probably rare and exotic, but new XAI techniques can make nearly all ML attacks easier and more damaging.
- Beware of insider threats, especially organized extortion of insiders.
- Open, public prediction APIs are a privacy and security nightmare.
- Your competitors could be gaming or stealing your public predictive models. Do your end user license agreements (EULA) or terms of service (TOS) explicitly prohibit this?
- Best practices around IT security, model management, and model monitoring are good defenses.

References

This presentation:

https://github.com/jphall663/secure_ML_ideas

Proposals for Model Vulnerability and Security:

https://www.oreilly.com/ideas/proposals-for-model-vulnerability-and-security

Can Your Machine Learning Model Be Hacked?!:
https://www.h2o.ai/blog/can-your-machine-learning-model-be-hacked/

References

[1] Julia Angwin et al. "Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks." In: *ProPublica* (2016). URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[2] Marco Barreno et al. "The Security of Machine Learning." In: *Machine Learning* 81.2 (2010). URL: https://people.eecs.berkeley.edu/~adj/publications/paper-files/SecML-MLJ2010.pdf, pp. 121–148.

[3] Anthony W. Flores, Kristin Bechtel, and Christopher T. Lowenkamp. "False Positives, False Negatives, and False Analyses: A Rejoinder to Machine Bias: There's Software Used across the Country to Predict Future Criminals. And It's Biased against Blacks." In: *Fed. Probation* 80 (2016). URL: https://bit.ly/2Gesf9Y, p. 38.

[4] Nicolas Papernot. "A Marauder's Map of Security and Privacy in Machine Learning: An overview of current and future research directions for making machine learning secure and private." In: *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*. URL: https://arxiv.org/pdf/1811.01134.pdf. ACM. 2018.

References

[5] Nicolas Papernot et al. "Scalable Private Learning with PATE." In: *arXiv preprint arXiv:1802.08908* (2018). URL: https://arxiv.org/pdf/1802.08908.pdf.

[6] Reza Shokri, Martin Strobel, and Yair Zick. "Privacy Risks of Explaining Machine Learning Models." In: *arXiv preprint arXiv:1907.00164* (2019). URL: https://arxiv.org/pdf/1907.00164.pdf.

[7] Reza Shokri et al. "Membership Inference Attacks Against Machine Learning Models." In: *2017 IEEE Symposium on Security and Privacy (SP)*. URL: https://arxiv.org/pdf/1610.05820.pdf. IEEE. 2017, pp. 3–18.

[8] Florian Tramèr et al. "Stealing Machine Learning Models via Prediction APIs." In: *25th {USENIX} Security Symposium ({USENIX} Security 16)*. URL: https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_tramer.pdf. 2016, pp. 601–618.

[9] Rich Zemel et al. "Learning Fair Representations." In: *International Conference on Machine Learning*. URL: http://proceedings.mlr.press/v28/zemel13.pdf. 2013, pp. 325–333.