Security for Machine Learning

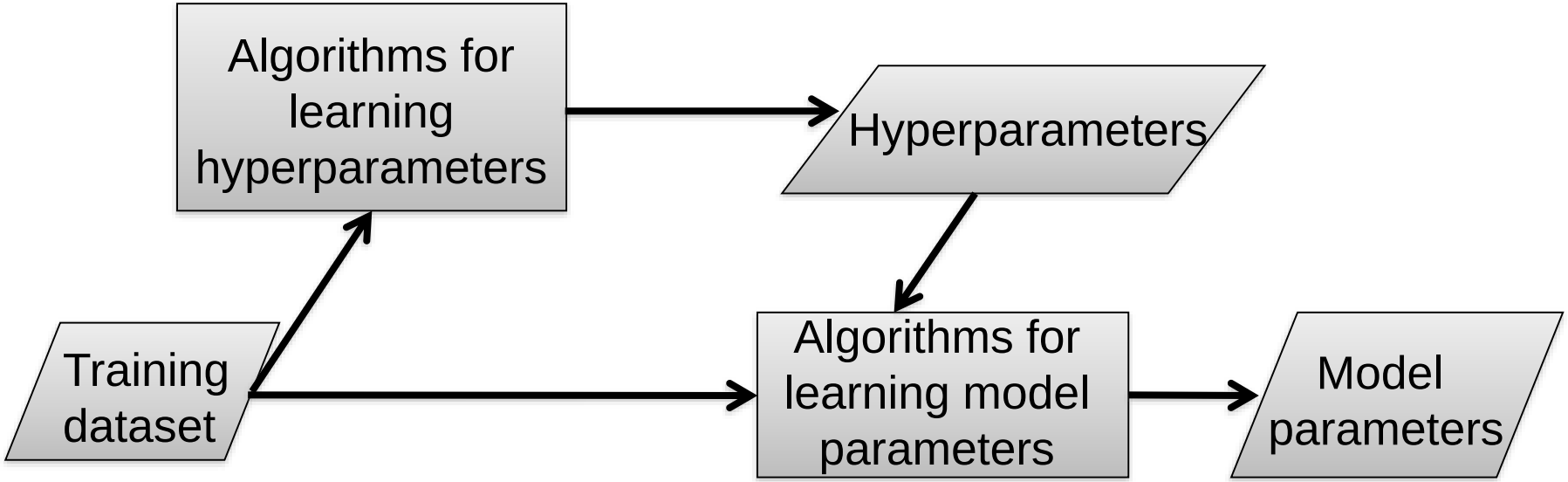
Neil Gong
ECE Department
Iowa State University
09/05/2018

2018 ISC 互联网安全大会 中国 · 北京
Internet Security Conference 2018 Beijing · China
（原中国互联网安全大会）

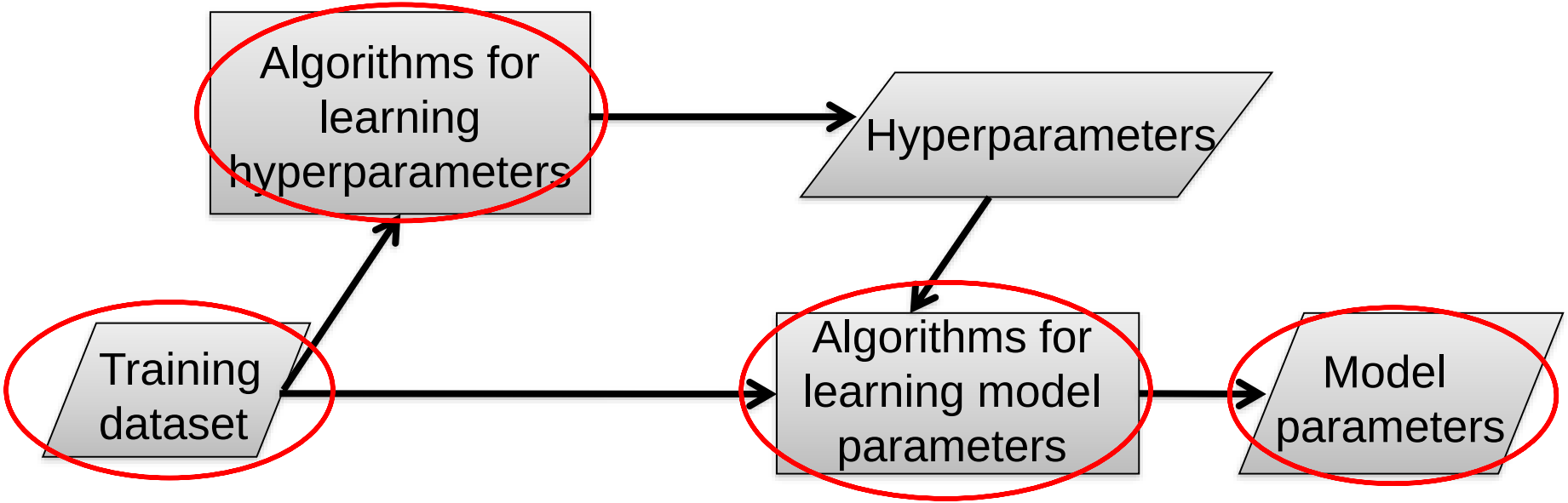
Security for Machine Learning

- Integrity
  - Training
  - Deployment/Prediction
- Confidentiality
  - Users: private training and testing data
  - Service providers: confidential algorithms, models, and hyperparameters

Training a Machine Learning Model



Compromising Integrity at Training



Approach:

Poisoning training data

Malicious algorithm

Goal:

Model with bad prediction accuracy

Trojaned model

Recommender Systems are Vulnerable to Training Data Poisoning Attacks

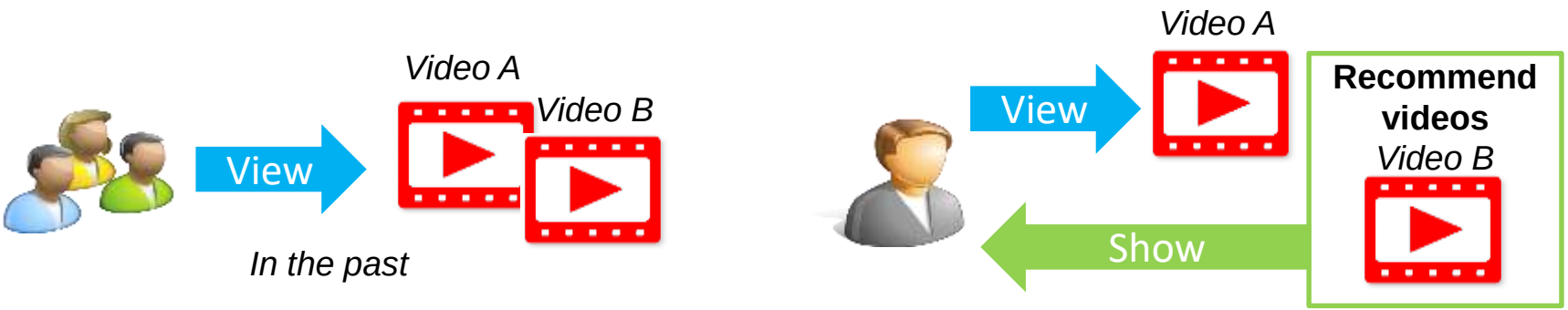
- Recommender system is an important component of Internet
  - Videos, products, news, etc.
- Common belief: recommend users items matching their interests
- Our work: injecting fake training data to make recommendations as an attacker desires

Guolei Yang, Neil Zhenqiang Gong, and Ying Cai. “Fake Co-visitation Injection Attacks to Recommender Systems”. In *NDSS*, 2017

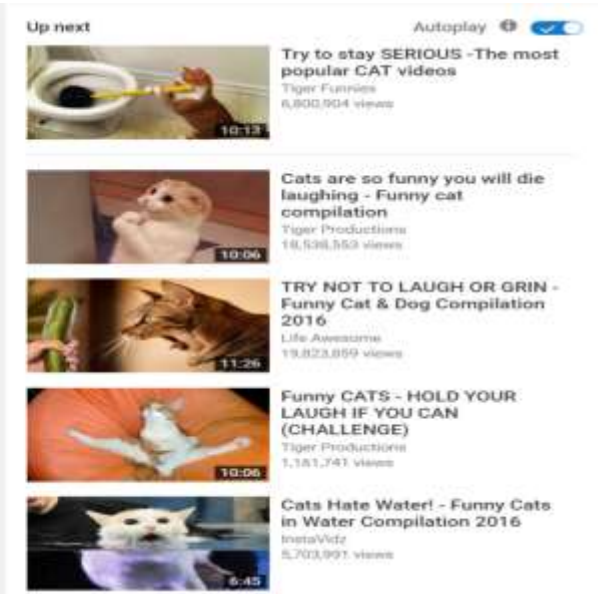
g, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. “Poisoning Attacks to Graph-mender Systems”. In *ACSAC*, 2018

Co-visitation Recommender Systems

- Key idea: Items that are frequently visited together in the past are likely to be visited together in the future



Co-visitation Recommender Systems



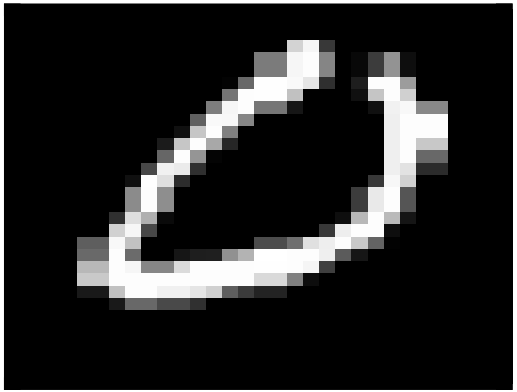
Our Attacks

- Goal: Promoting a target item
- Injecting fake co-visitations between a target item and some carefully selected items
  - The target item will appear in their recommendation lists
- Can attack YouTube, Amazon, eBay, LinkedIn, etc.

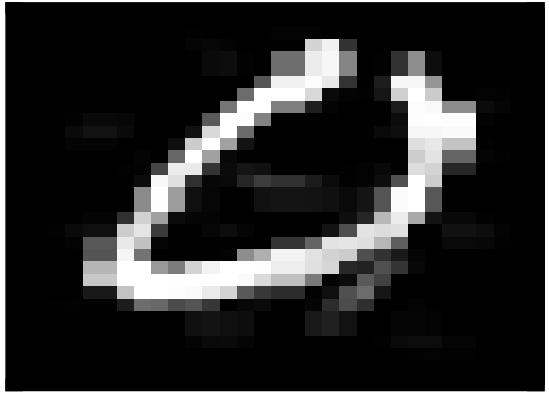
Security for Machine Learning

- Integrity
  - Training
  - **Deployment/Prediction: adversarial examples**
- Confidentiality
  - Users: private training and testing data
  - Service providers: confidential algorithms, models, and hyperparameters

Adversarial Examples



Normal example: digit 0



Adversarial example:
predicted to be 9

Adversarial Examples

- Normal example x
- Classifier C
- Adversarial example $x' = x + \delta$
- t: target label, $C(x') = t$

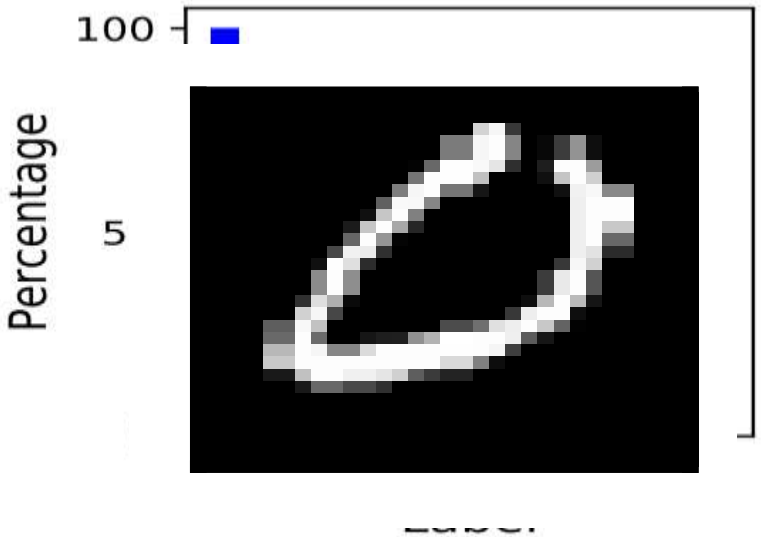
L_0, L_2, L_∞ norm of the noise δ

Minimize $d(x, x')$

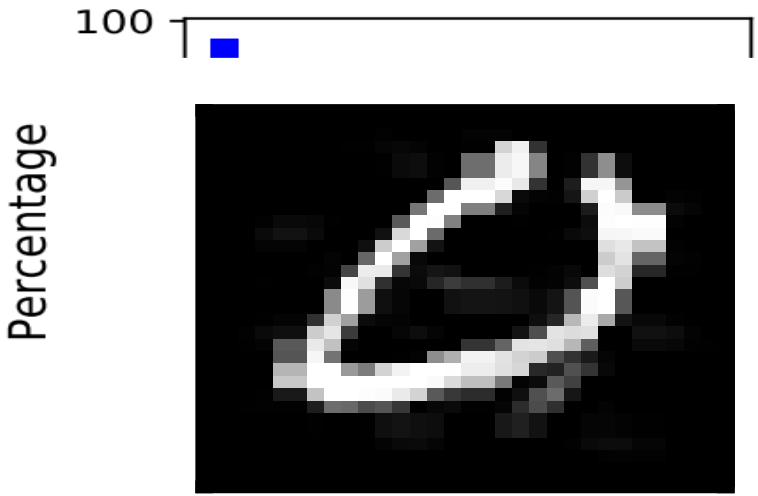
Subject to (1) $C(x') = t$

(2) *x' is a legitimate example*

Measuring Adversarial Examples



A normal example: digit 0



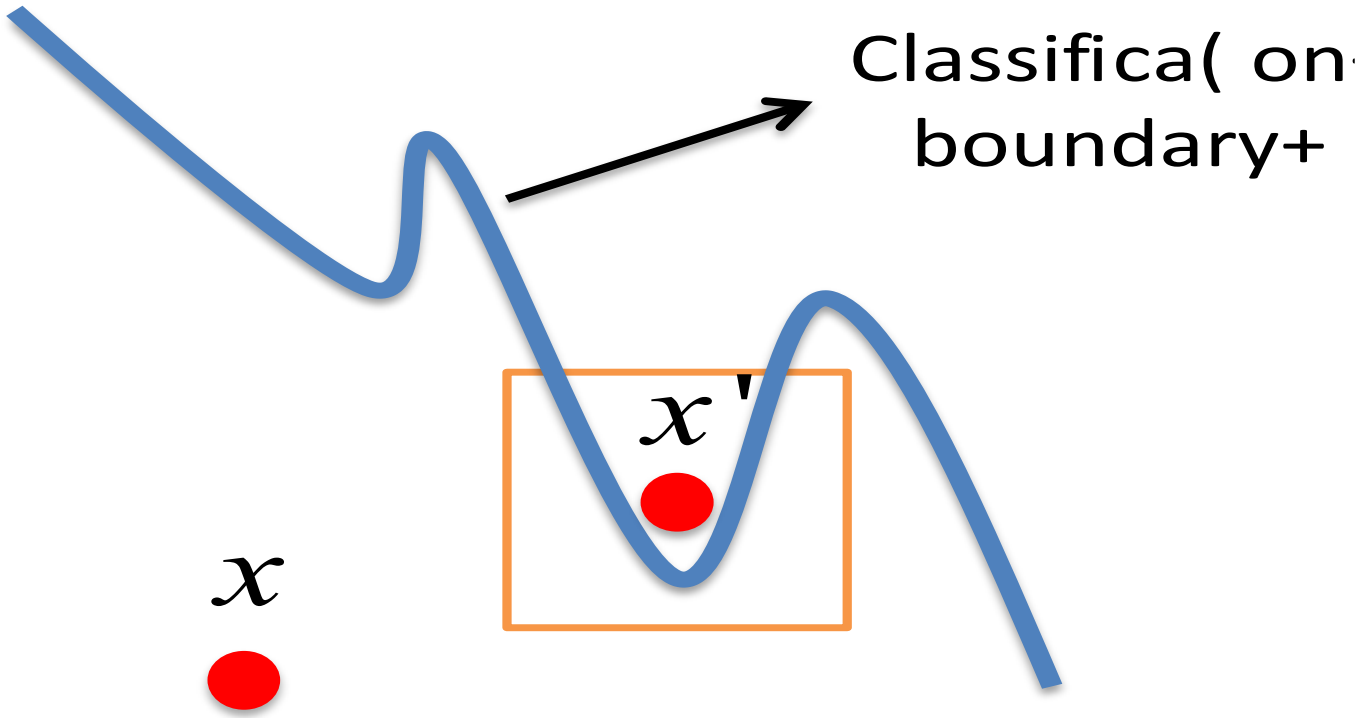
An adversarial example with a target label 9

Xiaoyu Cao and Neil Zhenqiang Gong. “Mitigating Evasion Attacks to Deep Neural gion-based Classification”. In *ACSAC*, 2017

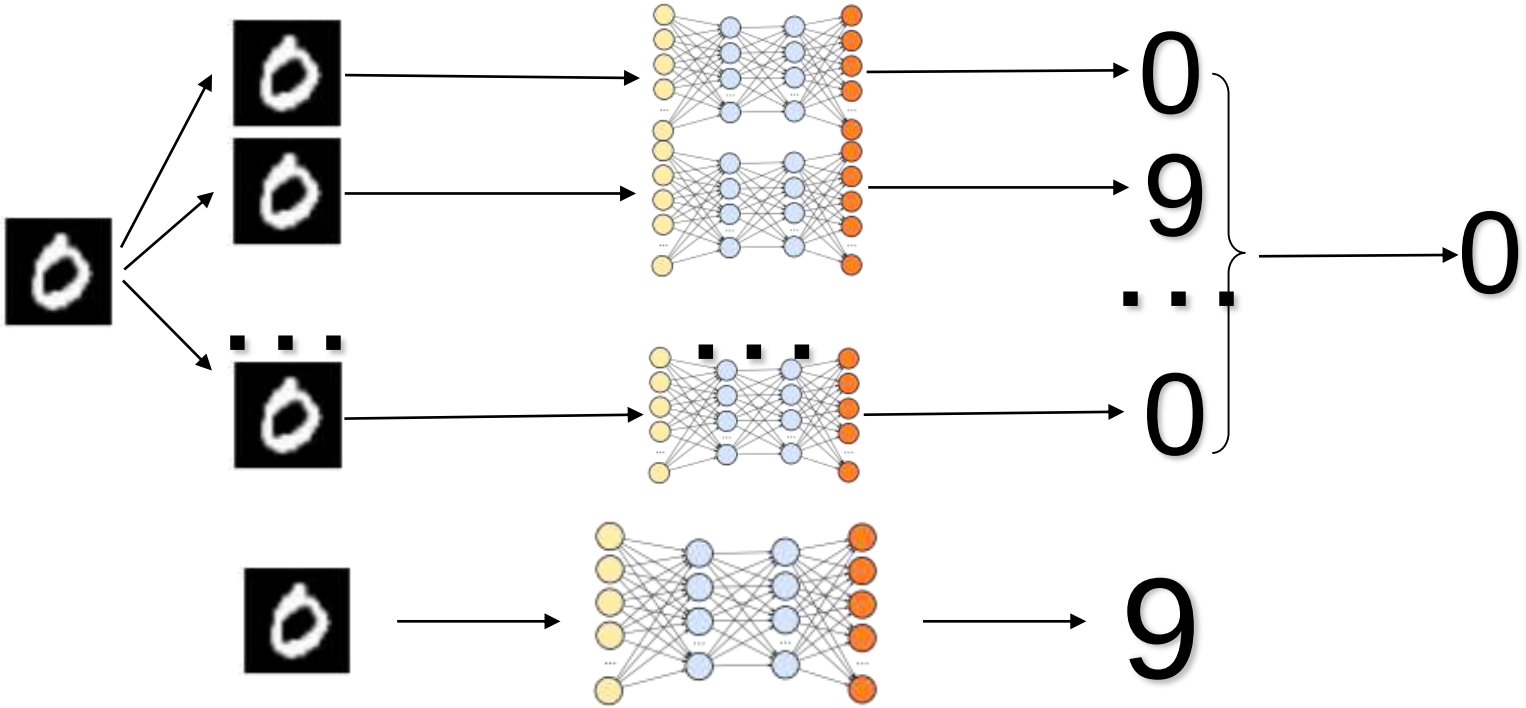
Observations

- Normal examples are not robust to small ***carefully crafted*** noise
  - Existence of adversarial examples
- Normal examples are robust to small ***random*** noise
- Adversarial examples are ***not*** robust to small ***random*** noise

Our Region-based Classification



Our Region-based Classification



Point-based classification

Evaluations on MNIST for Carlini and Wagner (CW) Attacks (IEEE S&P'17)

Different versions of CW attacks

Accuracy on normal examples

	Classification Accuracy	Success Rate		
		CW-L_0	CW-L_2	CW-L_∞
Standard point-based DNN	99.4%	100%	100%	100%
Adversarial training DNN	99.3%	100%	100%	100%
Distillation DNN	99.2%	100%	100%	100%
Our region-based DNN	99.4%	16%	0%	0%

Existing defenses

Mitigate adversarial examples without accuracy loss

Good Use of Adversarial Examples: Protecting Privacy

- Inference attacks: an attacker infers a user's private attributes using its public data
  - Private attributes: political view, sexual orientation, etc.
  - Public data: page likes on Facebook, rating scores, etc.
- An attacker has a classifier to infer private attributes
- A user's public data is a classification example

Good Use of Adversarial Examples: Protecting Privacy

- User adds *carefully crafted* noise to evade the attacker's classifier
  - Making the public data an "adversarial example"
- Key challenge: how to guarantee utility of the public data?

Jinyuan Jia and Neil Zhenqiang Gong. "AttriGuard: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning". In *Usenix Security*

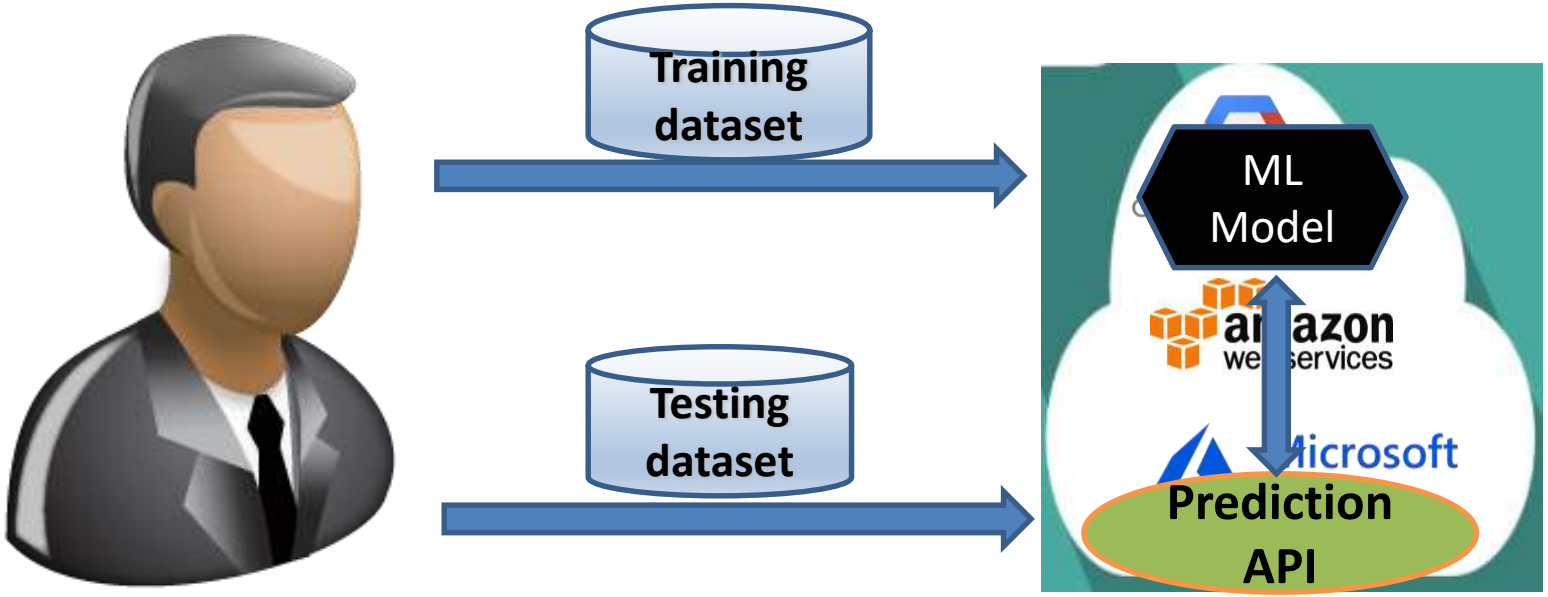
Security for Machine Learning

- Integrity
  - Training
  - Deployment/Prediction: adversarial examples
- **Confidentiality**
  - Users: private training and testing data
  - Service providers: confidential algorithms, models, and hyperparameters

Machine Learning as a Service (MLaaS)

- MLaaS enables users with limited computing power or limited machine learning expertise to use machine learning techniques

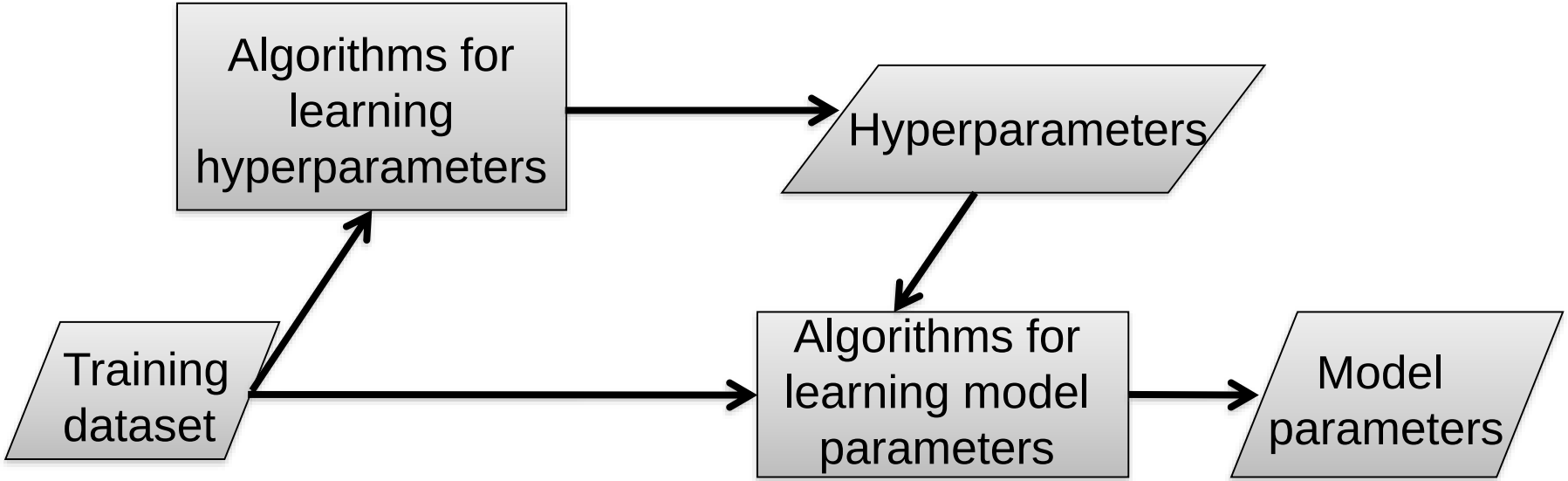
How MLaaS is Used?



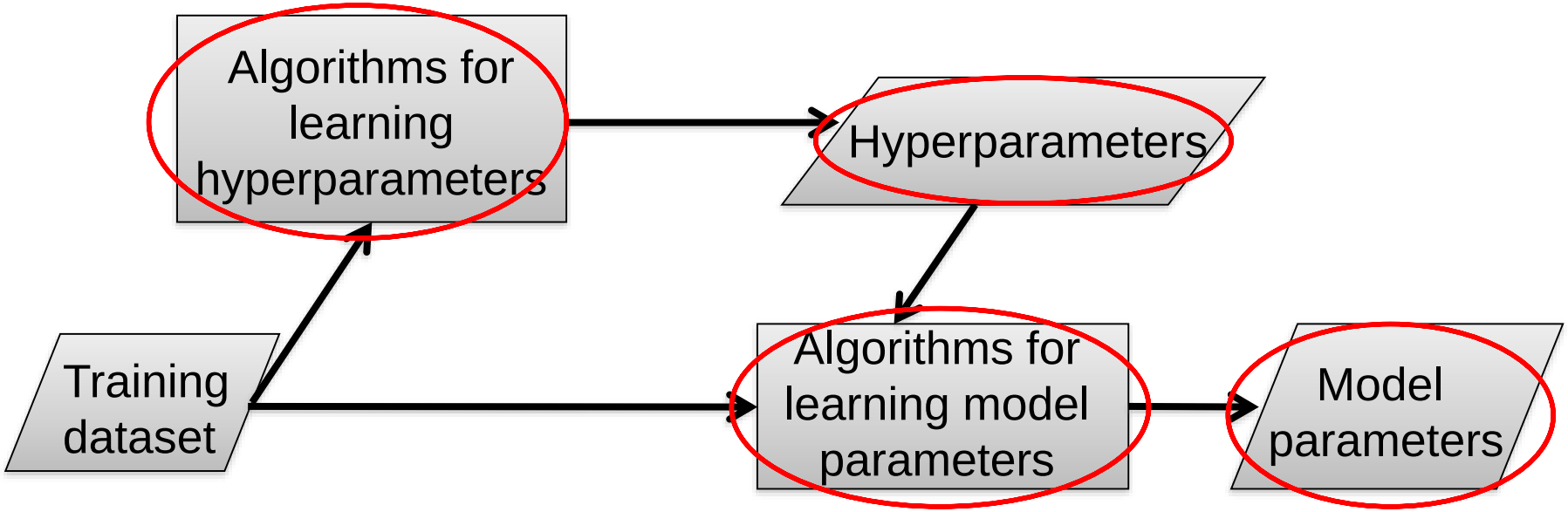
Confidentiality for Users

- Training data
- Testing data
- Approaches
  - Trusted processors, e.g., Intel SGX
  - Cryptographic techniques, e.g., secure multi-party computation
  - Statistical methods, e.g., differential privacy

Training a Machine Learning Model



Confidentiality for Service Providers



Stealing Hyperparameters

- We propose a general framework to steal hyperparameters in MLaaS
- Save economical costs without sacrificing model performance
- New defenses are needed

Binghui Wang and Neil Zhenqiang Gong. “Stealing Hyperparameters in Machine Learning”. In *IEEE Symposium on Security and Privacy*, 2018.

Conclusion

- Security is a big challenge for machine learning
- Integrity
  - Training
  - Deployment/Prediction
- Confidentiality
  - Users
  - Service providers