ECE 663: Machine Learning in Adversarial Settings

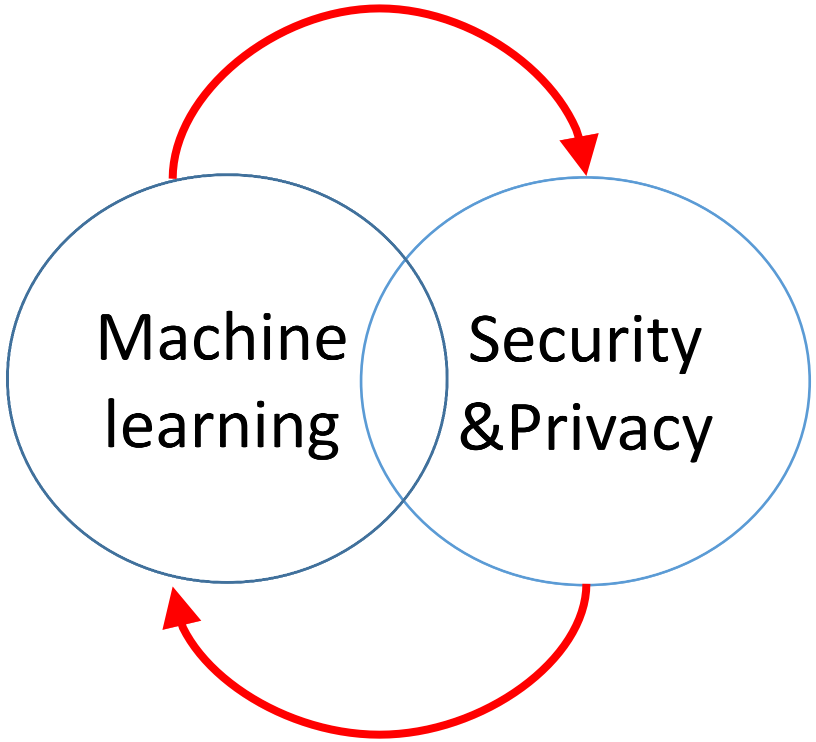
Neil Gong

Instructor

- Neil Gong
- neil.gong@duke.edu
- Research area
    - AI and security
- Office hour
    - Time: Thursday 9:00am – 10:00am
    - Location: 413 Wilkinson Building
- Teaching assistant
    - Zhengyuan Jiang, zhengyuan.jiang@duke.edu
    - Yuepeng Hu, yuepeng.hu@duke.edu

Course overview

- Security and privacy for machine learning
  - Security and privacy issues of ML
  - Secure and privacy-preserving ML
  - Beyond accuracy and efficiency of ML
- Machine learning for security and privacy
  - ML to enhance security
  - Misuse of ML



- Course webpage:

https://duke-ece663.github.io/Machine-Learning-in-Adversarial-Settings-Fall-2024/

Goal of this class

- State-of-the-art literature on adversarial machine learning
- Get prepared to apply and research adversarial machine learning

Class format

- Read papers
  - Write comments and send to adversarialmlduke@gmail.com
  - Deadline: Sunday and Tuesday 11:59pm
  - Send your comments to all papers in a single email thread
  - Comment
    - One paragraph of summary of each assigned paper
    - Three or more strengths
    - Three or more weaknesses
- Lead a lecture
  - Forming a group of at most 4 students
  - A group sends three preferred dates to adversarialmlduke@gmail.com by 11:59pm, 09/13
- Participate in class
- One class project
  - Can be a group of at most 4 students
  - Your research project can be class project
  - 09/16: project proposal due
  - 10/21: milestone report due
  - 11/20, 11/25: project presentation
  - 12/06: final project report due

Lead a lecture

- Why lead a lecture
  - Understanding a topic better after teaching others about it
- Like how I give a lecture
- May read multiple papers on the selected topic
  - E.g., each group member leads discussion on one paper
- 75 mins for a lecture!
- Use whiteboard/blackboard if possible
- Be interactive

An example class project

- Problem: finding adversarial examples in the white-box setting

- Solution: optimization-based method
  - E.g., start from the Carlini and Wagner method (to be discussed in the next lecture) as a baseline
  - Design a new method, e.g., enhance the Carlini and Wagner method via exploring new loss functions or use a different method to solve the formulated optimization problem

- Proposal abstract: one paragraph to describe the problem and potential solution.

Another example class project

- Problem: Detecting AI-generated images

- Solution: watermark
  - E.g., start from a watermarking method and optimize it to enhance its robustness, efficiency, and/or image quality

- Proposal abstract: one paragraph to describe the problem and potential solution.

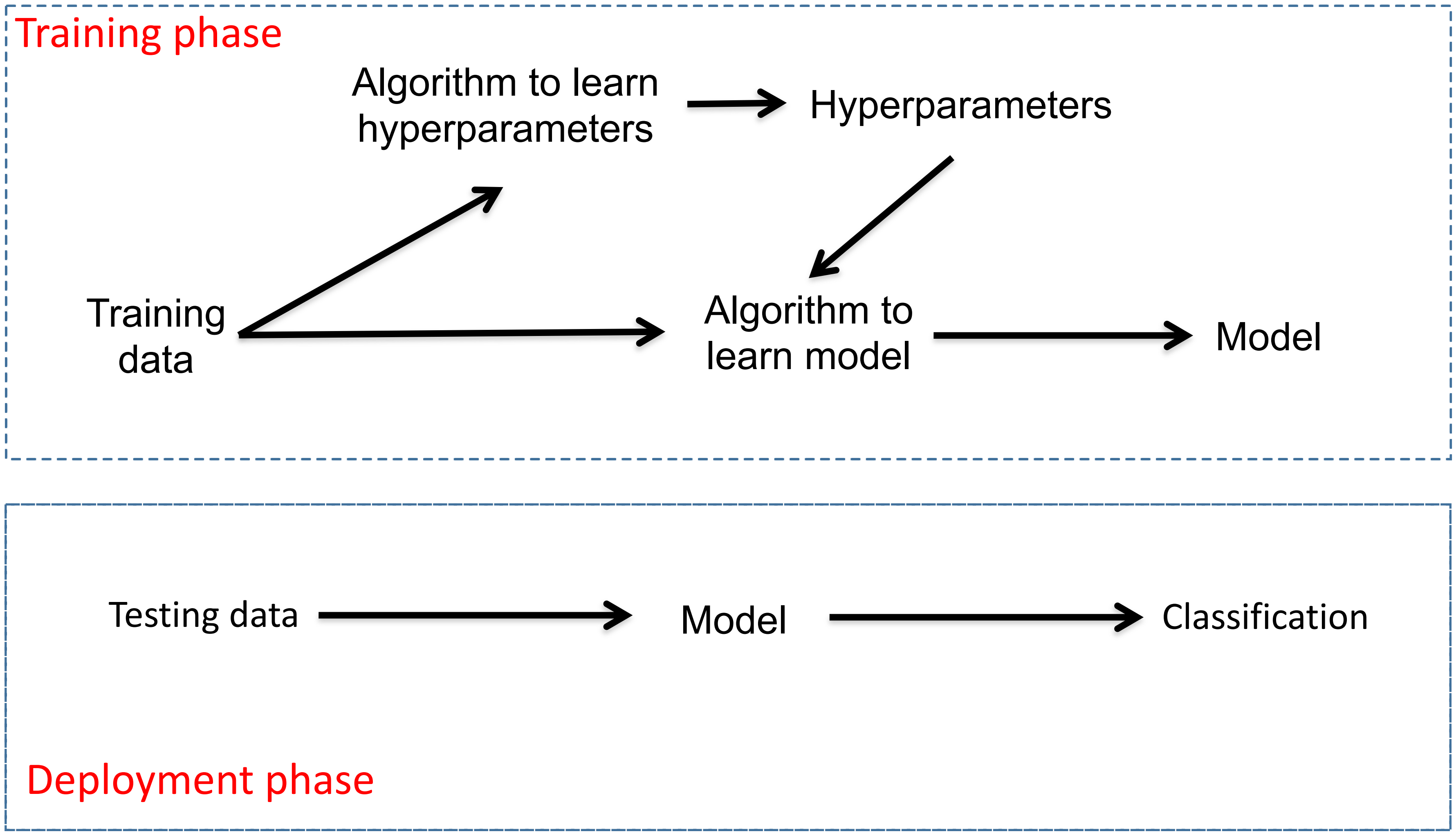
Project report template

- Abstract
- Introduction
- Related work (can also be moved to be after empirical evaluation)
- Problem definition
- Method
- Theoretical evaluation (if any)
- Empirical evaluation
- Conclusion

Grading policy

- 50% project
- 25% reading assignment
- 10% class participation
- 15% class presentation

Machine Learning Pipeline

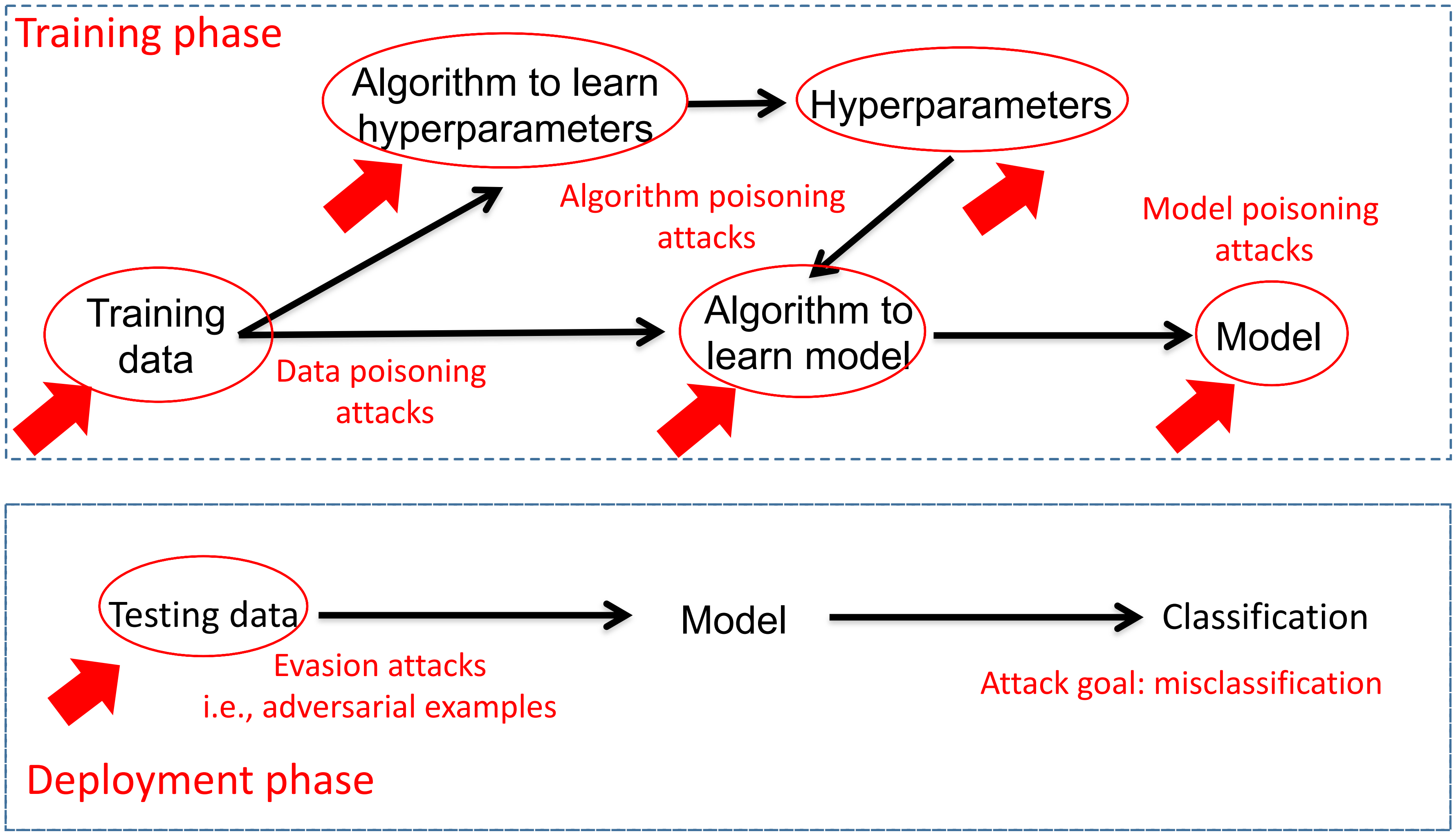


Security of Machine Learning

- Integrity
  - Training phase
  - Deployment phase

- Confidentiality
  - Training/testing data
  - Model parameters
  - Hyperparameters
  - Algorithms

Integrity of Machine Learning



Attack Goal: Misclassification

- Untargeted
  - Arbitrary misclassification

- Targeted
  - Attacker-chosen misclassification

Poisoning Attacks – Attack Training Phase

- Compromise training phase to poison the learnt model
- Poisoned model misclassifies testing inputs as attacker desires

- Data poisoning
  - Modify training data to poison the model

- Algorithm poisoning
  - Modify algorithm to poison the model
  - E.g., when ML library is from untrusted third party

- Model poisoning
  - Directly modify parameters of the model
  - E.g., model is from third party or model training is distributed (federated learning)

An Example of Data Poisoning Attack

(x_1, y_1)
(x_2, y_2) $\longrightarrow$ f
(x_3, y_3)

$x \xrightarrow{f} y$

No data poisoning

(x_1, y_1)
(x_2, y_2)
(x_3, y_3) $\longrightarrow$ f'
(x_4, y_4)

$x \xrightarrow{f'} y'$

Injecting carefully crafted training data

Evasion Attacks – Attack Deployment Phase

- Model is clean
- Attacker perturbs testing inputs to induce misclassification

Classification:
Panda

+

Carefully crafted
perturbation
(physically realizable)

=

Adversarial
example

Classification:
Monkey

Defenses to Protect Integrity

- Empirically secure defenses
  - Secure against specific, known attacks
  - Vulnerable to advanced, adaptive attacks

- Provably secure defenses
  - Secure against arbitrary attacks satisfying certain constraints
  - Often sacrifice accuracy when no attacks

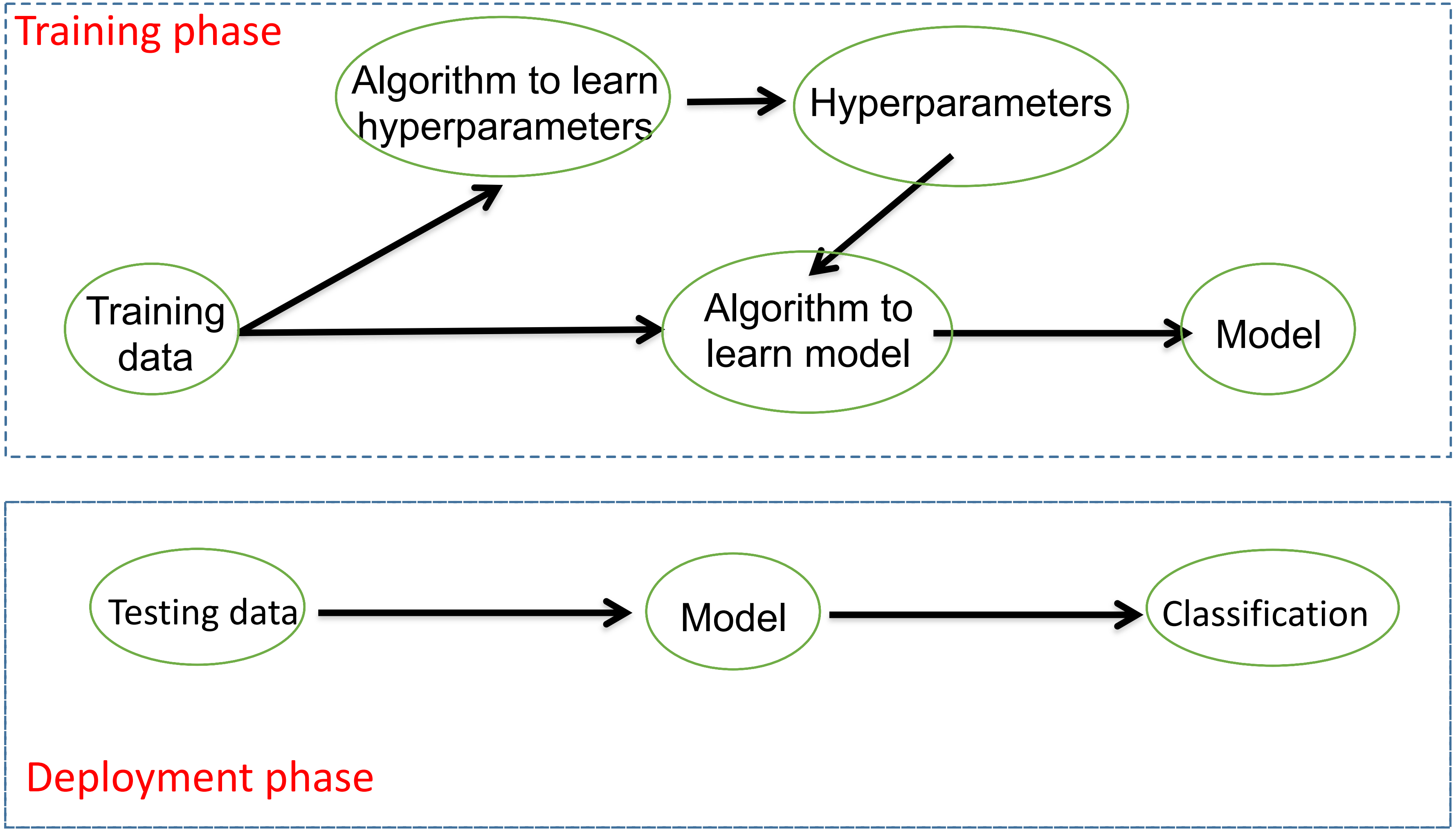
- Still open challenges

Defenses against Evasion Attacks

- Adversarial training (empirically secure defense)
  - Key idea: use adversarial examples with correct labels to augment training data

- Randomized smoothing (provably secure defense)
  - Key idea: add random noise to a testing input to overwhelm adversarial perturbation (if any) before classifying it
  - Predicted label is unaffected by arbitrary adversarial perturbation whose L_p norm is bounded

Confidentiality of Machine Learning



Attacks to Confidentiality of Machine Learning

- Model stealing
  - Reconstruct a model's exact parameters or learn a functionality-equivalent surrogate one via querying the model
- Hyperparameter stealing
  - Reconstruct hyperparameters used to train a model
- Membership inference
  - Infer whether a given input is in a given model's training data
- Training data reconstruction (also known as model inversion)
  - Reconstruct training data of a given model

Summary

- Course overview