

## Adversarial Machine Learning II

### IMPORTANT NOTES:

**Study lecture materials at least 1 hour and prepare the questions prior to the tutorial session. The questions will be discussed in the tutorial session.**

1. Outline the primary stages of a typical classification task. Discuss the importance of each stage in achieving accurate pattern recognition.

The primary stages of a classification task are:

- Pre-processing: Align, normalize, or standardize input data (e.g., resizing images) to ensure consistency. This stage is critical for reducing noise and variability that could degrade model performance.
- Feature Extraction: Identify and extract salient features (e.g., edges in images, keywords in text) that distinguish classes. This step reduces dimensionality and focuses the model on relevant patterns.
- Classification: Compare extracted features against learned patterns using a distance metric (e.g., Euclidean distance) or probabilistic model (e.g., neural networks). This stage directly determines the predicted class label.

2. Consider a critical infrastructure company that outsources the training of its AI model, which is designed to detect anomalies in sensor data. Describe how a malicious actor could leverage a BadNet attack in this scenario. What immediate and long-term consequences could such an attack have on the infrastructure?

A malicious actor outsources training and injects a backdoor (e.g., specific sensor readings trigger misclassification as "normal"). The model behaves correctly on clean data but fails when the trigger (e.g., a unique noise pattern) appears. This results in anomalies (e.g., equipment failures) go undetected, causing operational disruptions or safety hazards. In long-term, this causes the erosion of trust in AI systems, financial losses from downtime, and potential physical damage to infrastructure.

3. Differentiate between an "Untargetted attack" and a "Targetted attack" within the context of changing labels. Provide a unique real-world example for each type of attack.

- Untargetted Attack: Randomly misclassify backdoored samples (e.g., change any "cat" image to "dog"). For example, a spam filter mislabels 10% of emails randomly, causing both false positives and negatives.
- Targeted Attack: Misclassify specific samples to a chosen class (e.g., all "stop signs" → "speed limit"). For example, an autonomous vehicle misinterprets red traffic lights as green, risking collisions.

4. If a training dataset contains 50,000 samples and an attacker decides to poison 0.5% of these samples with a backdoor, how many samples would be affected? Additionally, explain how an attacker might manipulate parameters like "step size" (learning rate), "batch size," and "epochs" to minimize their cost function during this attack.

If the training dataset contains 50,000 samples, the affected samples is 250 (0.5% of 50,000). To minimize the cost function during this attack, the attacker can manipulate the following parameters:

- Step Size (Learning Rate): Smaller steps avoid overshooting minima but may slow convergence; larger steps risk instability.

- Batch Size: Smaller batches update weights more frequently (noisier but precise), while larger batches stabilize gradients.
  - Epochs: More epochs improve backdoor integration but risk overfitting. The attacker balances these to minimize detection while ensuring trigger effectiveness.
5. Compare and contrast the adversarial capabilities and limitations of an attacker in BadNet versus TrojanNet. Highlight the key differences in their access to training data and model parameters.
- Training data access: BadNet requires poisoning the training data while TrojanNet has no access to the training data and can only insert triggers post-training.
  - Model modification: BadNet alters weights via poisoned data while TrojanNet adds Trojan module without changing the weights.
  - Defense difficulty: BadNet is detectable through data auditing while TrojanNet is harder to detect since there is no data tampering.
6. Imagine you are a product manager at a company that develops autonomous vehicles, and your team is considering purchasing pre-trained machine learning models from a third-party vendor. What specific security risks would you be concerned about, and what initial verification steps would you propose before integrating such models into your vehicles?

The models may be vulnerable to backdoor attacks. To assess their robustness against such threats, Universal Litmus Patterns (ULPs) can be used to detect anomalous behavior. Techniques like randomized smoothing and the use of adversarial samples can further evaluate the models' resilience. As a preventive measure, a denoiser (Blackbox Smoothing) can also be trained to sanitize inputs before classification.