HANDOUT 6 - WEEK 7 READING

Semi-Supervised and Reinforcement Learning

CIS 350: AI for Cybersecurity - Smart Defense for the Digital Business

Colorado State University - Computer Information Systems

How to use this reading

This handout supports Week 7 and completes the four styles of machine learning. It covers SEMI-SUPERVISED learning (learning from a few labels plus lots of unlabeled data) and REINFORCEMENT learning (learning a strategy from rewards), and how both apply to security - from malware detection to automated, adaptive defense. The figures and code match Lab 6; the review questions map to Quiz 6.

Estimated reading time: 20-25 minutes. No prior coding background needed.

1. Completing the Map: Four Ways Machines Learn

You have met supervised learning (labeled examples) and unsupervised learning (no labels). This week fills in the rest. SEMI-SUPERVISED learning sits between them: a few labels plus lots of unlabeled data. REINFORCEMENT learning is different in kind - it learns from experience and rewards rather than a fixed dataset. Together these four paradigms cover how nearly every AI system learns.

The four paradigms in one line

Supervised: learn from labeled data. Unsupervised: find structure with no labels. Semi-supervised: a few labels + lots of unlabeled data. Reinforcement: learn a policy from rewards by interacting. Real systems often combine several.

2. Semi-Supervised Learning

Labels are expensive - confirming malware, fraud, or intrusions takes scarce expert time - while unlabeled data (files, flows, logs) is essentially free and unlimited. Semi-supervised learning uses BOTH: the few labels anchor the classes, and the many unlabeled points reveal their shape. The key assumption is that similar (nearby) points tend to share a label, so labels can spread from labeled points to similar unlabeled ones.

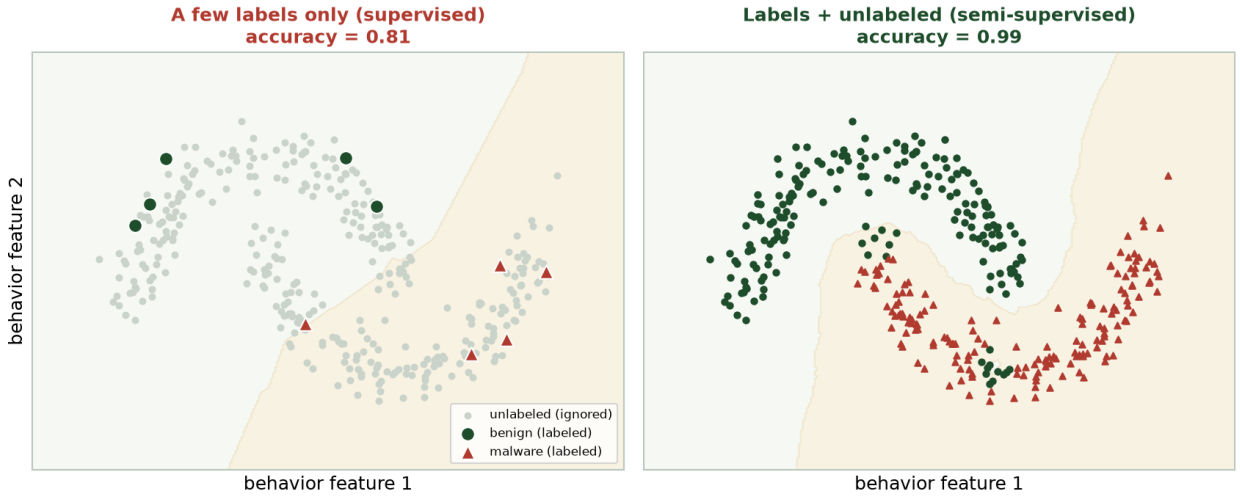

Figure 1. Left: 10 labels only - a supervised model draws the wrong boundary (81%). Right: adding the unlabeled data recovers the two interleaving groups (99%). Same 10 labels.

- Self-training: train on the few labels, add the most confident predictions as new labels, repeat.
- Label spreading/propagation: labels flow along a graph connecting similar points.
- In scikit-learn, mark unlabeled rows with -1 so the model learns their labels from structure.
- Caution: it relies on the structure assumption - confidently-wrong guesses can spread errors.

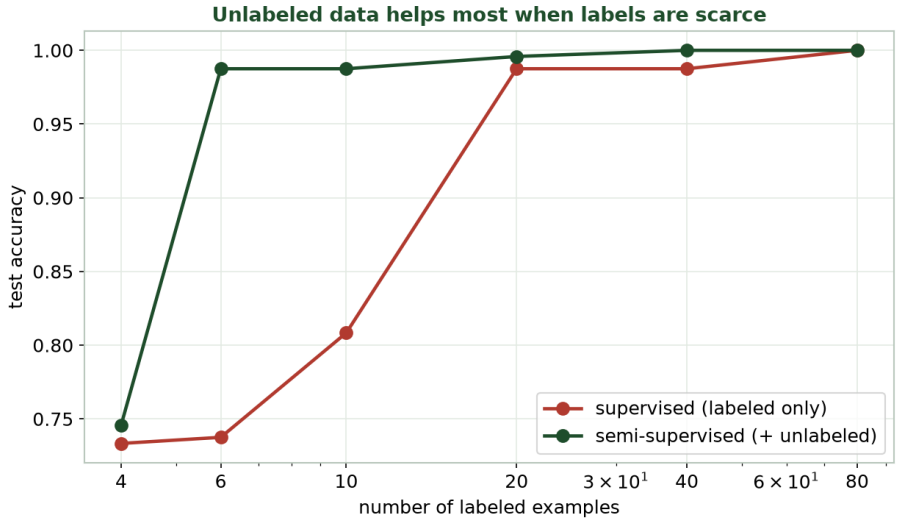

Figure 2. Unlabeled data helps most when labels are scarce; the gap shrinks as labels grow. Security usually lives on the left (few labels).

3. Reinforcement Learning

Reinforcement learning (RL) has no dataset of right answers. An AGENT takes ACTIONS in an ENVIRONMENT; each action returns a new state and a REWARD (or penalty). The agent's goal is to earn the most reward over time, and what it learns is a POLICY - a strategy for choosing actions in each situation. It improves by trial and error over many attempts.

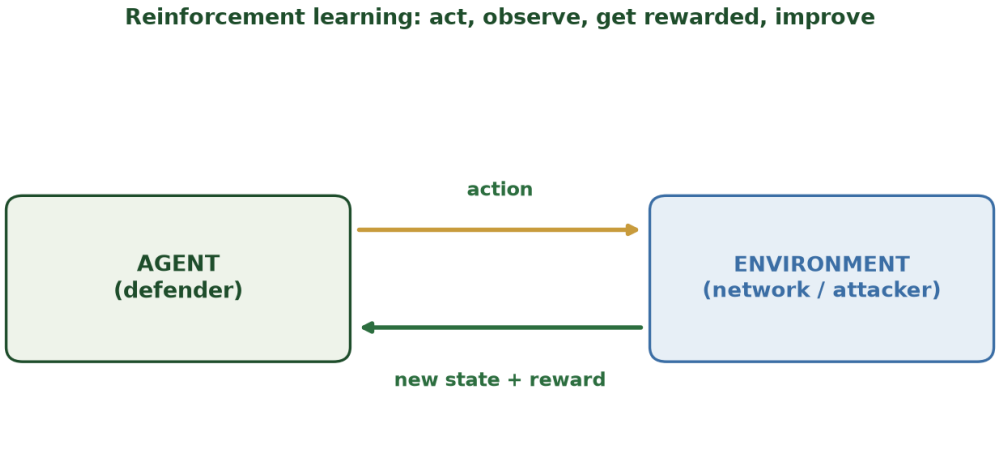

Figure 3. The RL loop: the agent acts, the environment returns a new state and reward, and the agent updates its policy.

- Policy: the learned strategy (what to do in each state).
- Reward shaping: good outcomes earn points, bad outcomes lose them - designing rewards well is an art.
- Exploration vs. exploitation: try new actions vs. use what already works.
- RL fits sequential decisions where each action changes the situation - like defending a network.

4. Security as a Defender-Attacker Game

Security is not a static dataset - the adversary reacts. That makes defense a GAME: each side's outcome depends on the other's choices, and a fixed defense decays as attackers adapt. RL lets a defender LEARN a strong, adaptive policy by playing many rounds against attacker behavior.

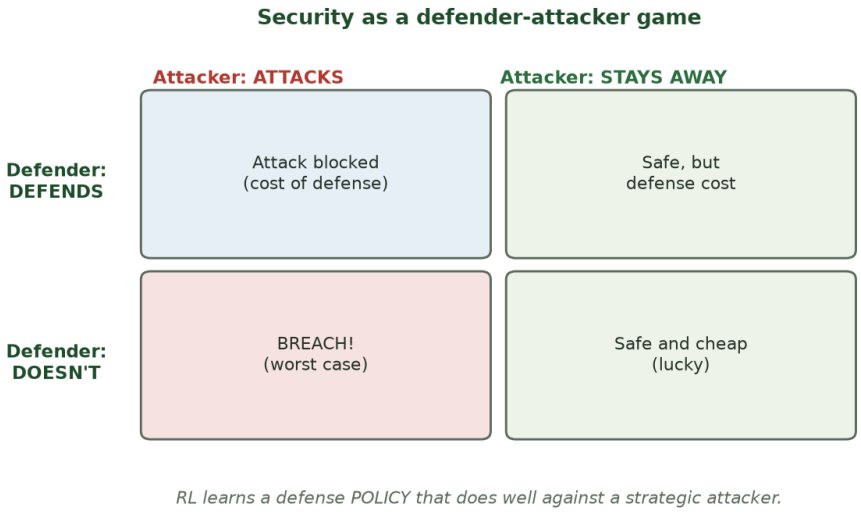

Figure 4. A defender-attacker game. The best move depends on the adversary, so an adaptive learned policy beats a fixed rule.

Automate responsibly

RL can power automated response (isolate a host, throttle traffic) faster than humans and adapt as attacks change. But a bad or manipulated policy can disrupt the business, and 'reward hacking' lets an agent game a poorly designed reward. Train in simulators/honeynets, and keep a human in the loop for high-impact actions.

5. The Code (Lab 6 Preview)

Semi-supervised learning is a few lines in scikit-learn - the one new idea is the -1 label for unlabeled rows. The RL loop is shown as pseudocode (RL security agents train in simulators, not live production):

```
# Semi-supervised: mark unlabeled rows -1, then let labels spread
from sklearn.semi_supervised import LabelSpreading
y_semi = y.copy(); y_semi[unlabeled] = -1
model = LabelSpreading(kernel='knn', n_neighbors=10).fit(X, y_semi)

# Reinforcement learning loop (concept only)
state = env.reset()
for step in range(many_episodes):
    action = agent.choose(state)            # e.g., isolate host? block IP?
    state, reward, done = env.step(action)   # attacker + network respond
    agent.learn(state, action, reward)       # update the defense policy
```

Key Terms

- Semi-supervised learning - learning from a few labels + lots of unlabeled data.
- Label spreading / self-training - ways to grow labels using data structure.
- -1 label - the scikit-learn convention marking an unlabeled row.
- Reinforcement learning - learning a policy from rewards by interacting.
- Agent, environment, action, state, reward - the RL building blocks.
- Policy - the agent's strategy for choosing actions.
- Exploration vs. exploitation - trying new actions vs. using known-good ones.
- Defender-attacker game - a strategic, adaptive security contest.
- Reward hacking - an agent exploiting a badly designed reward.

Review Questions (self-check for Quiz 6 and Lab 6)

- What is semi-supervised learning, and when does it help most?
- How does the -1 label convention work?
- What does a reinforcement-learning agent learn, and from what?
- Explain exploration vs. exploitation in your own words.
- Why is security described as a defender-attacker game?
- Give one benefit and one risk of automating defense with RL.

Remember: AI tools are not permitted on quizzes and labs in this course.

References

- scikit-learn User Guide - Semi-Supervised Learning (LabelSpreading, SelfTrainingClassifier).
- R. S. Sutton & A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. (free online).
- NIST, Artificial Intelligence Risk Management Framework (AI RMF 1.0), 2023.
- MITRE ATT&CK and MITRE D3FEND - attacker/defender technique knowledge bases.
- Clarence Chio & David Freeman, Machine Learning and Security (O'Reilly).
- Google, Machine Learning Crash Course (developers.google.com/machine-learning).