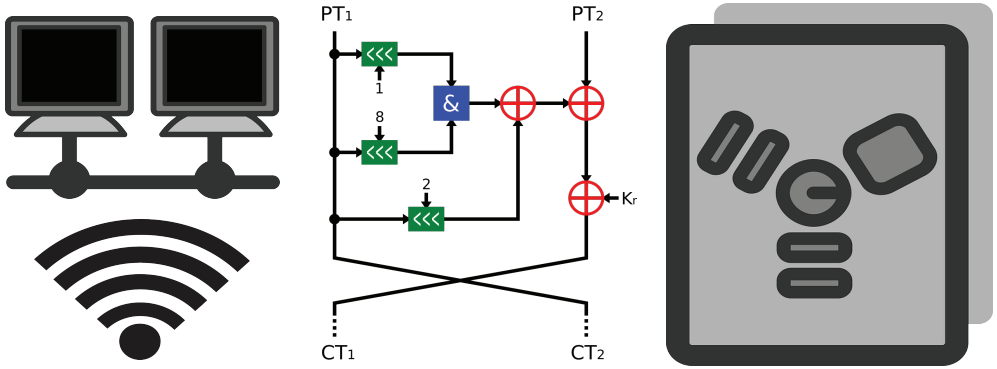
CSE 40567 / 60567: Computer Security



Network Security 6

Homework #7 is Due **tonight**
at 11:59PM (your timezone)

See **Assignments Page** on the course website for details

Final Exam Plan:

- Final will be released at 2pm on 5/7
- You will have 24hrs to complete it (due 2pm on 5/8)
- Open book / notes / Internet
- Format is short answer
- See the topic checklist on the course website

Course Instructor Feedback (CIF)

Deadline: 11:59PM, 5/3/20

***Being used this semester to assess online learning

General limitations of intrusion detection

1. In the general case, detecting intrusions reduces to a computationally hard problem
    - (Cohen 1994) proved that virus detection is as hard as the halting problem
    - We will never achieve a perfect IDS solution
2. Some attacks generate errors, and others do not
    - The latter case is problematic
3. Reactive IDS complicates policies regarding network traffic
    - Opens the door to DoS
    - May break normal use with unanticipated actions

General limitations of intrusion detection

4. High-cost of false alarms
    - Check your snort logs — the system cries wolf more often than you'd expect.
5. Policy problem: redlining
    - Certain netblocks are routinely flagged (China, Russia)
    - This can lead to unintended discrimination

Specific problems detecting network attacks

```
TCP 60 45680 > 50624 [ACK] Seq=1347 Ack=214330
Win=64975 Len=0 [Malformed Packet]
```

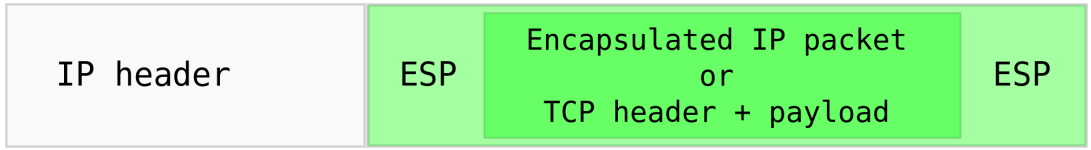
- The Internet is a very noisy environment
  - Many (most?) malformed packets are the result of software bugs
  - Drives up the false positive rate of the IDS
- There are very few real attacks **?**
  - 10 real attacks per million with a false alarm rate of 0.1%. What is the ratio of false to real alarms?

100:1

Specific problems detecting network attacks

```
$ wc -l emerging-all.rules.txt
73884 emerging-all.rules.txt
```

- Many attacks are specific to particular versions of software
  - IDS needs a large and constantly changing library of attack signatures



- Encrypted traffic is a stumbling block
  - IPSec protects both headers and payloads

Evading IDS

Example: polymorphic shellcode

Static shellcode is trivial to detect via a signature:

```
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
  "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
  "\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

Solution for attacker: encrypt the shellcode

Polymorphic shellcode routine

FAKENOP:

Generate two-byte instructions, the second byte of which is a one-byte instruction or the first byte of a two-byte instruction

```
\x15\x11\xF8\xFA\x81\xF9\x27\x2F\x90\x9E
```

Start at first byte

```
ADC $0x11F8FA81
STC
DAA
DAS
NOP
SAHF
```

Or

Start at second byte

```
ADC %eax,%edx
CMP %ecx,$0x272F909E
```

Polymorphic shellcode routine

DecipherRoutine:

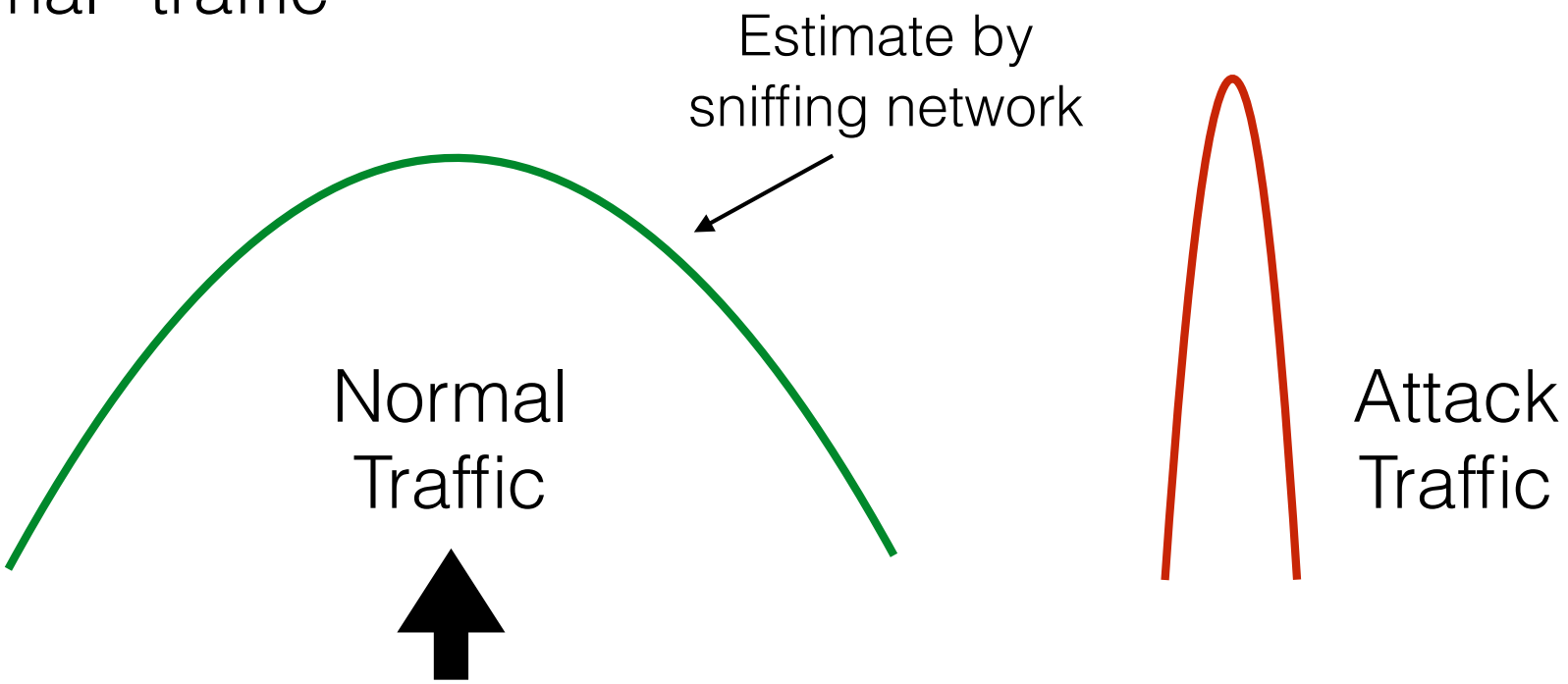
This can't be the same between attacks. 2 strategies:

1. ~~Use the same routine, but change the instructions~~
2. Generate different routines for decipher

- Generate routines which cipher with several instructions XOR, ADD, ROR
- Use random registers
- Four byte encryption

Polymorphic shellcode routine

Ideal strategy: make encrypted shellcode resemble "normal" traffic



Distribution of encrypted shellcode bytes should be consistent with this distribution

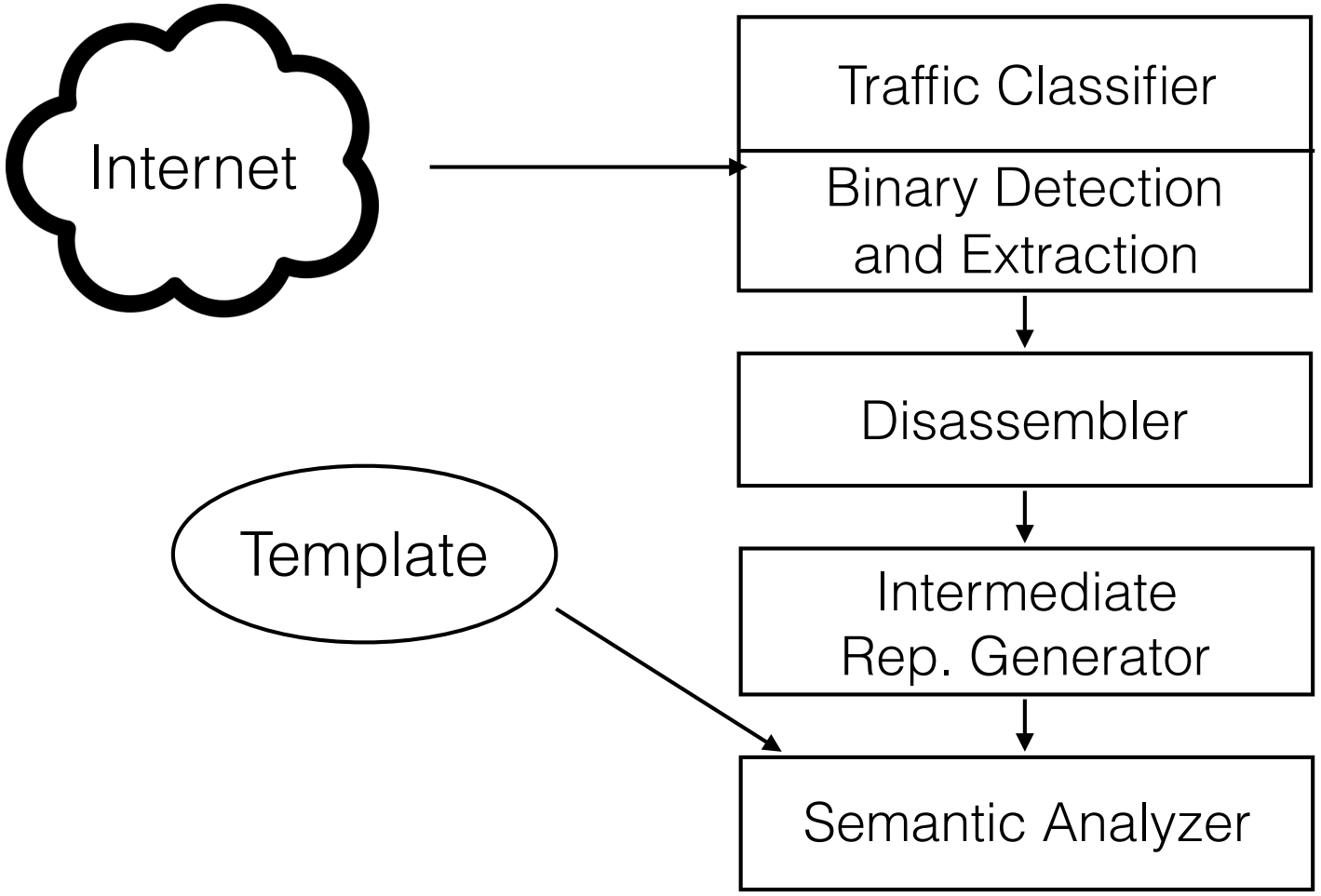
Polymorphic shellcode routine

What about the NOPs and distributional sampling?

- Sample these instructions so that the distribution is also consistent with normal traffic
- Problem: the set of instructions is smaller than the set of all the hex codes in the network traffic

Semantics-aware IDS

Strategy: don't look at the code, instead, consider its behavior

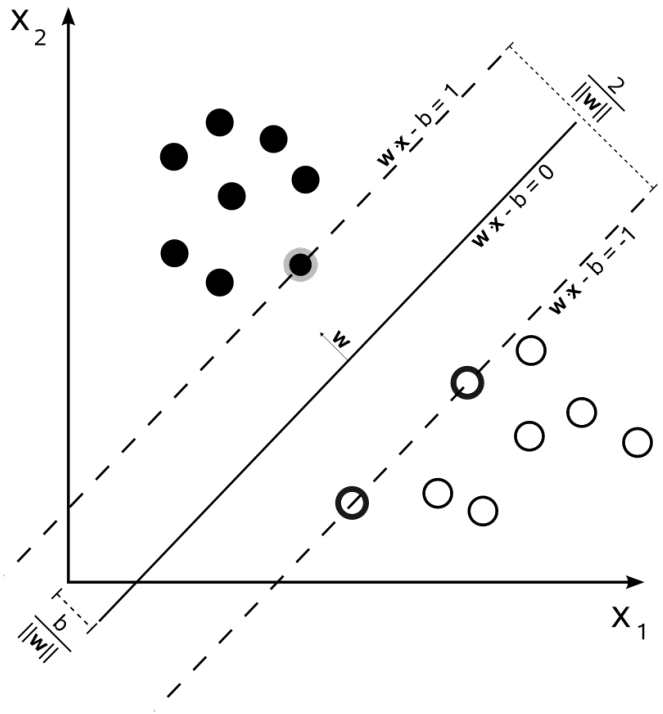


Machine Learning

People are good at reading logs, computers are not

- Can we teach machines to read logs like humans?

Machine Learning: learning and making predictions from data without explicit programming



Anomaly Detection IDS

Assumption: Attacks exhibit characteristics that are different than those of normal traffic

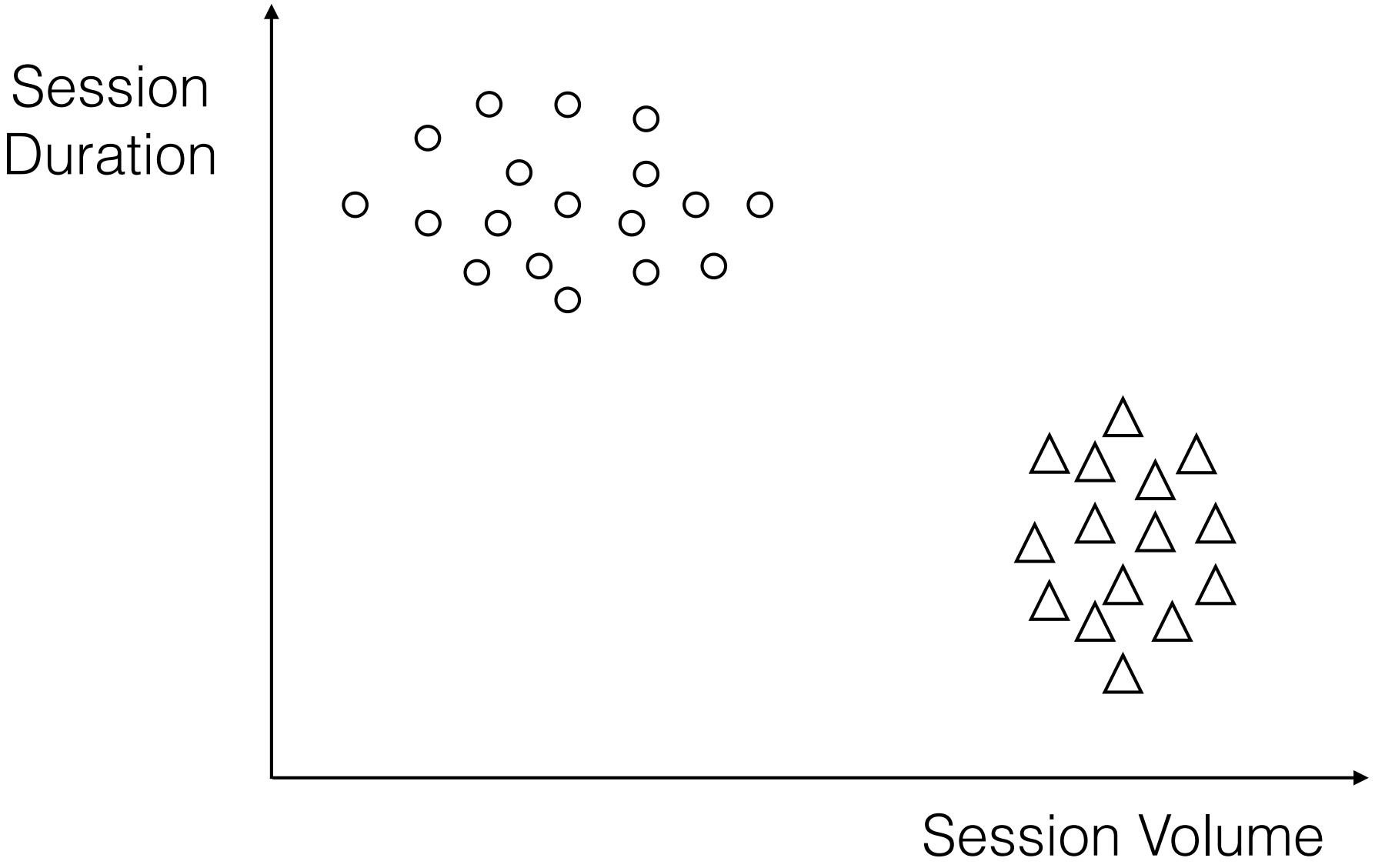
Promises to find novel attacks without anticipating specifics

Machine learning works well in other domains (e.g., computer vision, natural language processing). Why not IDS?

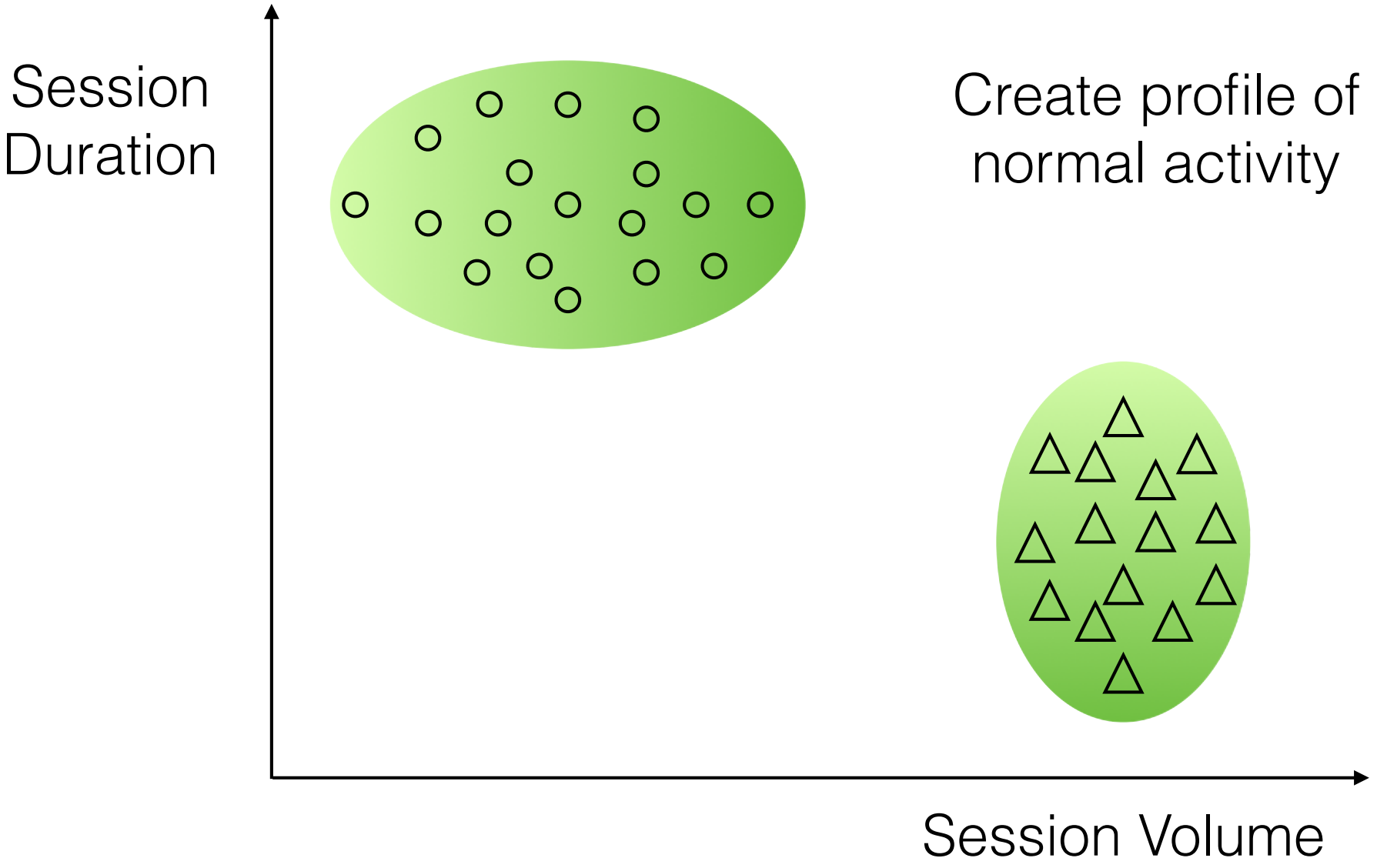
Rarely deployed in 2019

R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," 2010

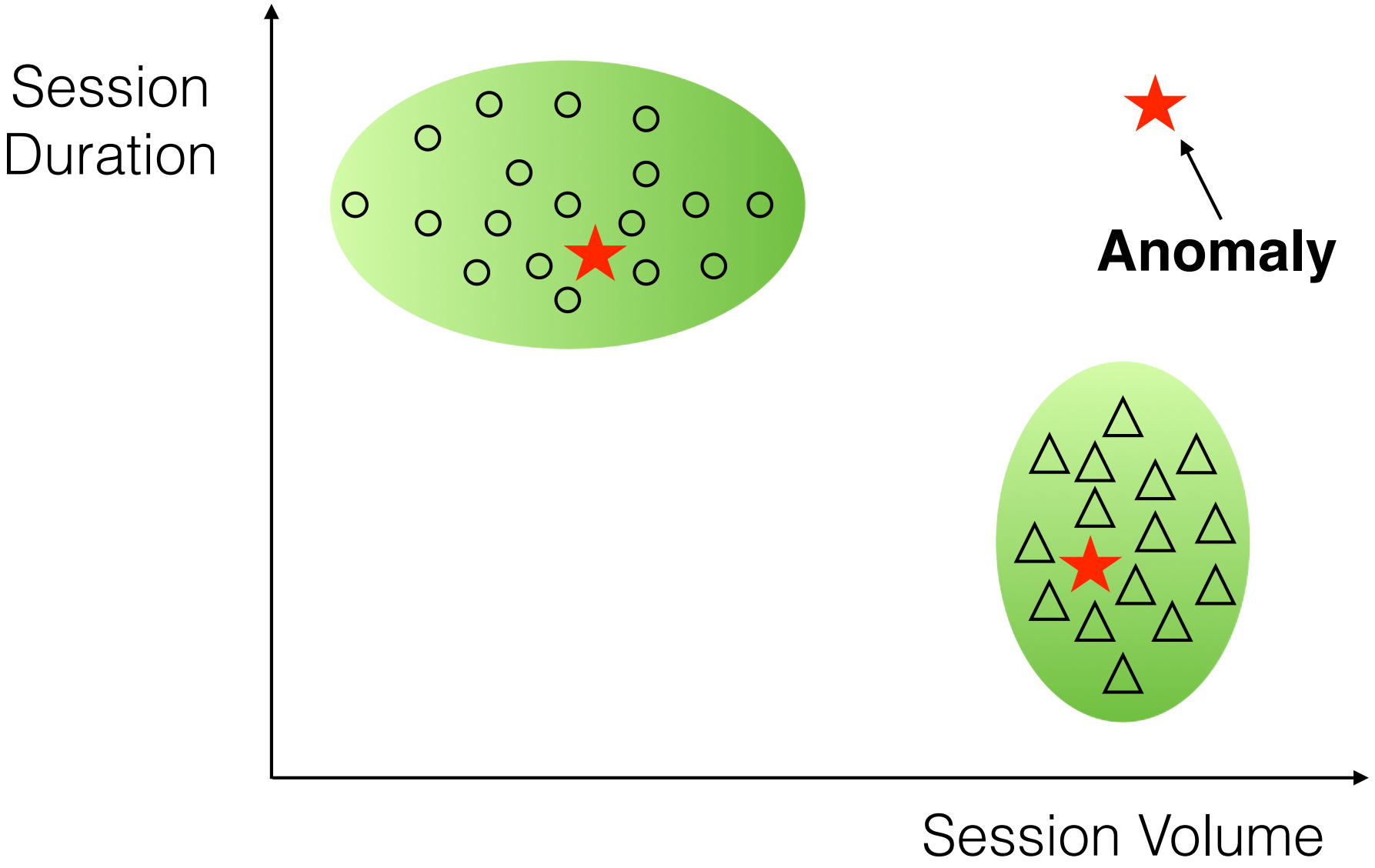
Anomaly Detection IDS: Data Space



Anomaly Detection IDS: Training



Anomaly Detection IDS: Testing

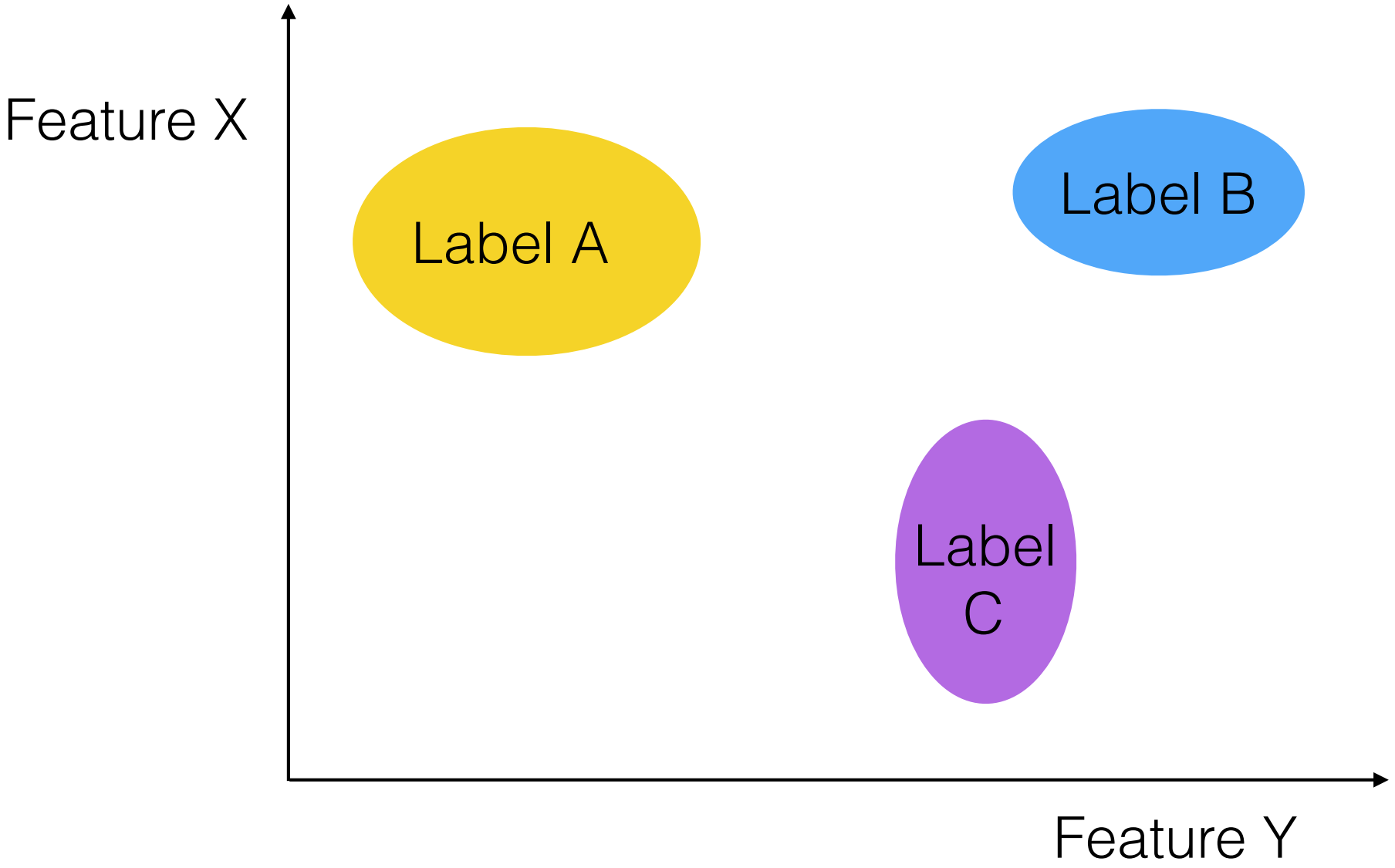


The difficulty of anomaly detection

Significant differences from other problem domains exist:

- How do we find the opposite of normal?
- What is the cause of the anomaly?
- How can we make sure it works?
- What is the feature space?
- Can the attacker fool the machine learning?

Feature space



Labeled Anomaly space