Poison Forensics: Traceback of Data Poisoning Attacks in Neural Networks

Shawn Shan, Arjun Nitin Bhagoji, Haitao Zheng, Ben Y. Zhao
U. Chicago

Discussion Lead: Evan Rose

Forensics Background

What is Forensics?

forensic 2 of 2 **noun**

1 **:** an argumentative exercise

2 **forensics** **plural in form but singular or plural in construction** **:** the art or study of argumentative discourse

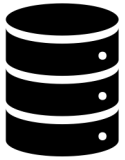
3 **forensics** **plural in form but singular or plural in construction** **:** the application of scientific knowledge to legal problems

especially **:** scientific analysis of physical evidence (as from a crime scene)

Source: Merriam-Webster

Forensics in Computer Science

- **Digital Forensics**: the collection, preservation, and analysis of digital data and actions in a manner that is admissible in court
- **Digital Forensics and Incident Response (DFIR)**: identifying and responding to cybersecurity incidents



Why do Forensics?

Accountability for responsible parties

Recovery from attacks

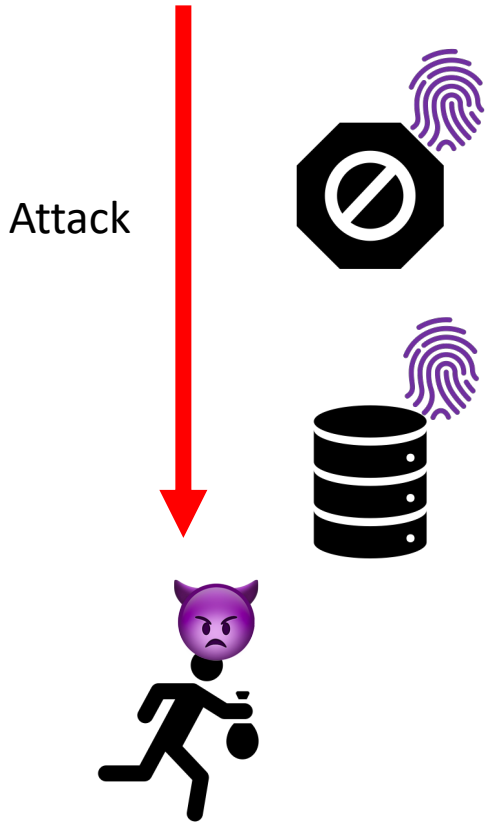
Forensics in Computer Science

- **Digital Forensics**: the collection, preservation, and analysis of digital data and actions in a manner that is admissible in court
- **Digital Forensics and Incident Response (DFIR)**: identifying and responding to cybersecurity incidents

Why do Forensics?

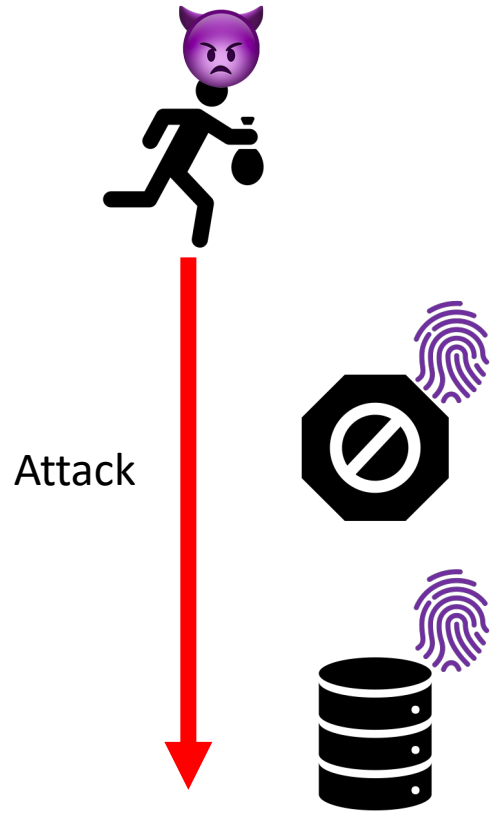
Accountability for responsible parties

Recovery from attacks



Forensics in Computer Science

- **Digital Forensics**: the collection, preservation, and analysis of digital data and actions in a manner that is admissible in court
- **Digital Forensics and Incident Response (DFIR)**: identifying and responding to cybersecurity incidents



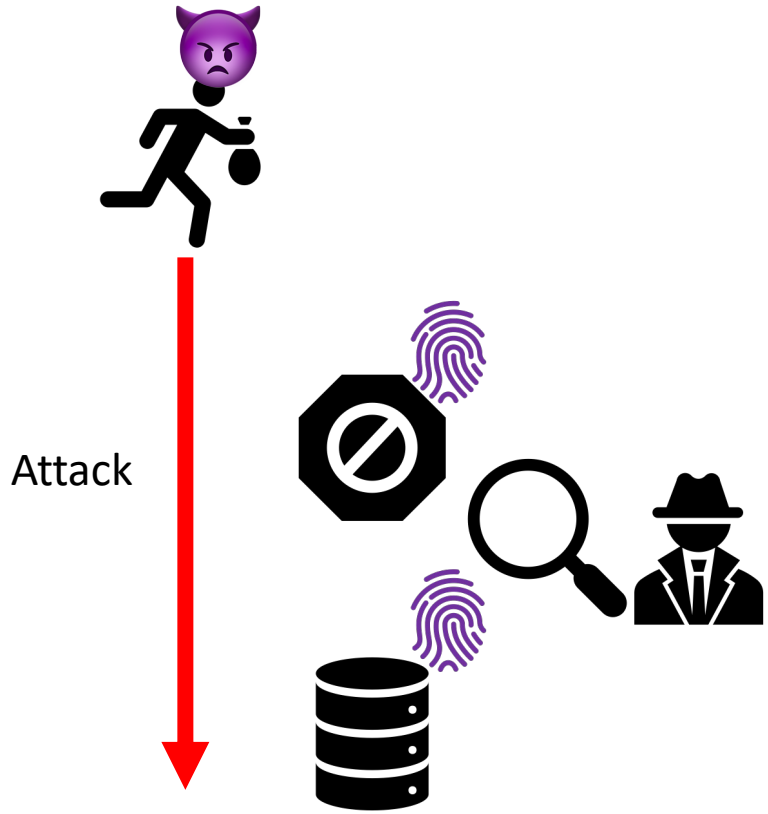
Why do Forensics?

Accountability for responsible parties

Recovery from attacks

Forensics in Computer Science

- **Digital Forensics**: the collection, preservation, and analysis of digital data and actions in a manner that is admissible in court
- **Digital Forensics and Incident Response (DFIR)**: identifying and responding to cybersecurity incidents



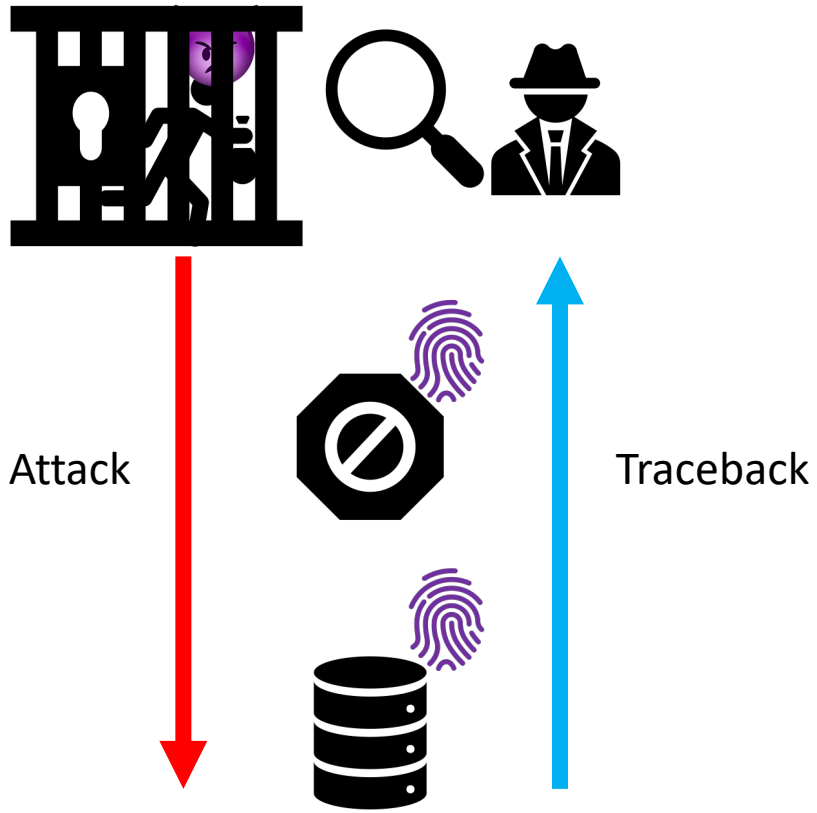
Why do Forensics?

Accountability for responsible parties

Recovery from attacks

Forensics in Computer Science

- **Digital Forensics**: the collection, preservation, and analysis of digital data and actions in a manner that is admissible in court
- **Digital Forensics and Incident Response (DFIR)**: identifying and responding to cybersecurity incidents



Why do Forensics?

Accountability for responsible parties

Recovery from attacks

Forensics for ML

- Specifically, **data poisoning**
- Given:
  - Model F trained on training data D
  - Misclassification event (x_a, y_a)

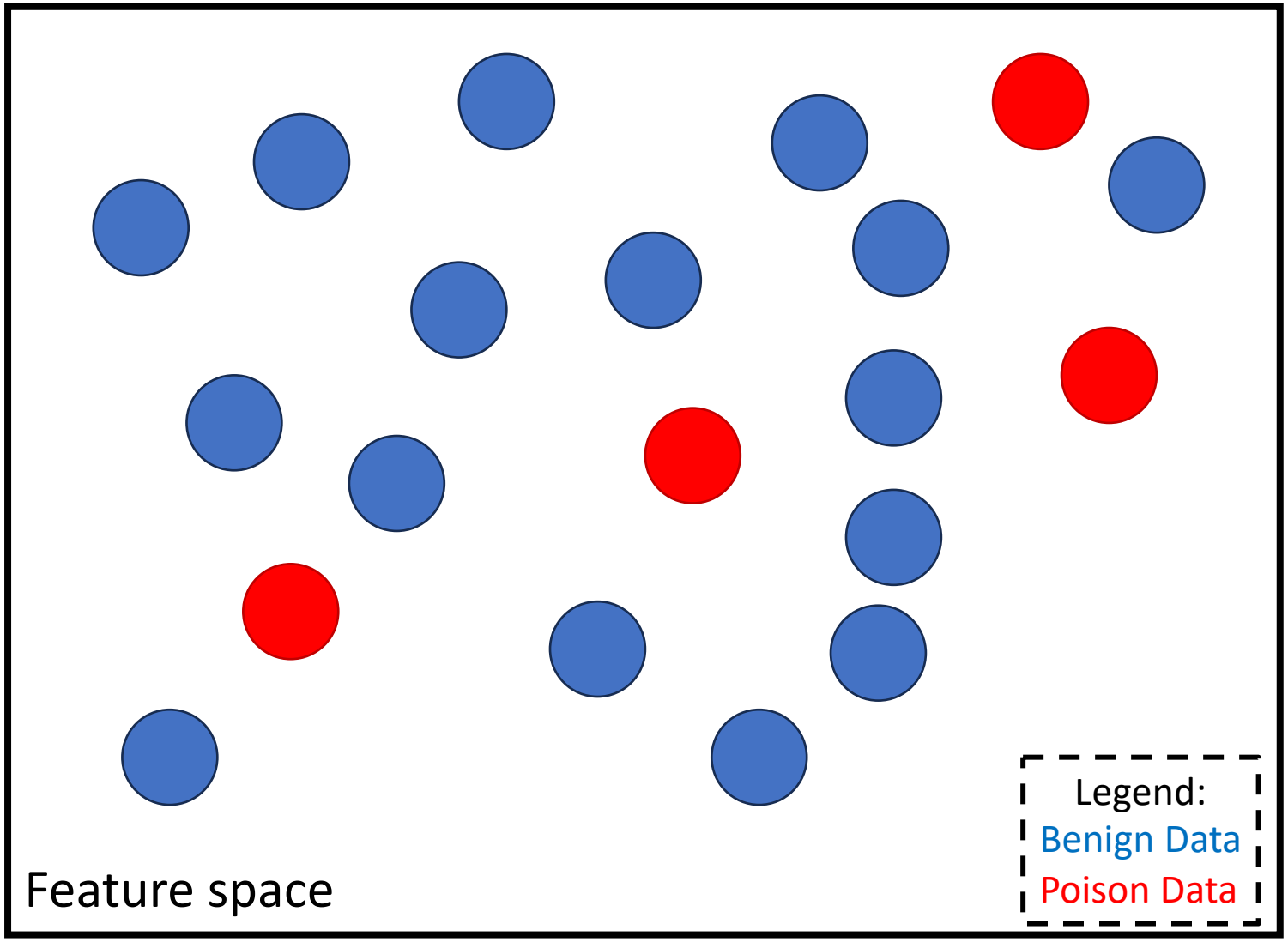
- Goal:
  - Determine a set of **poisons** $S \subseteq D$ responsible for the event
- Desired properties:
  - High **precision** (negative consequences for accused parties)
  - High **recall** (system effectiveness)
  - **Generality** across attack types

Defenses vs. Forensics

- **Defense**: stop the attacker **before** the model is trained / deployed
  - E.g., by some kind of outlier detection
  - Hard to anticipate attack strategies in advance!
  - Preferable to prevent damage
- **Forensics**: detect and respond to attacks when defenses **fail**
  - Damage is already done
  - But, able to leverage new knowledge about attacker
- Modern security (outside of ML) uses **both** forms of defense

Core Mechanism: Cluster and Prune

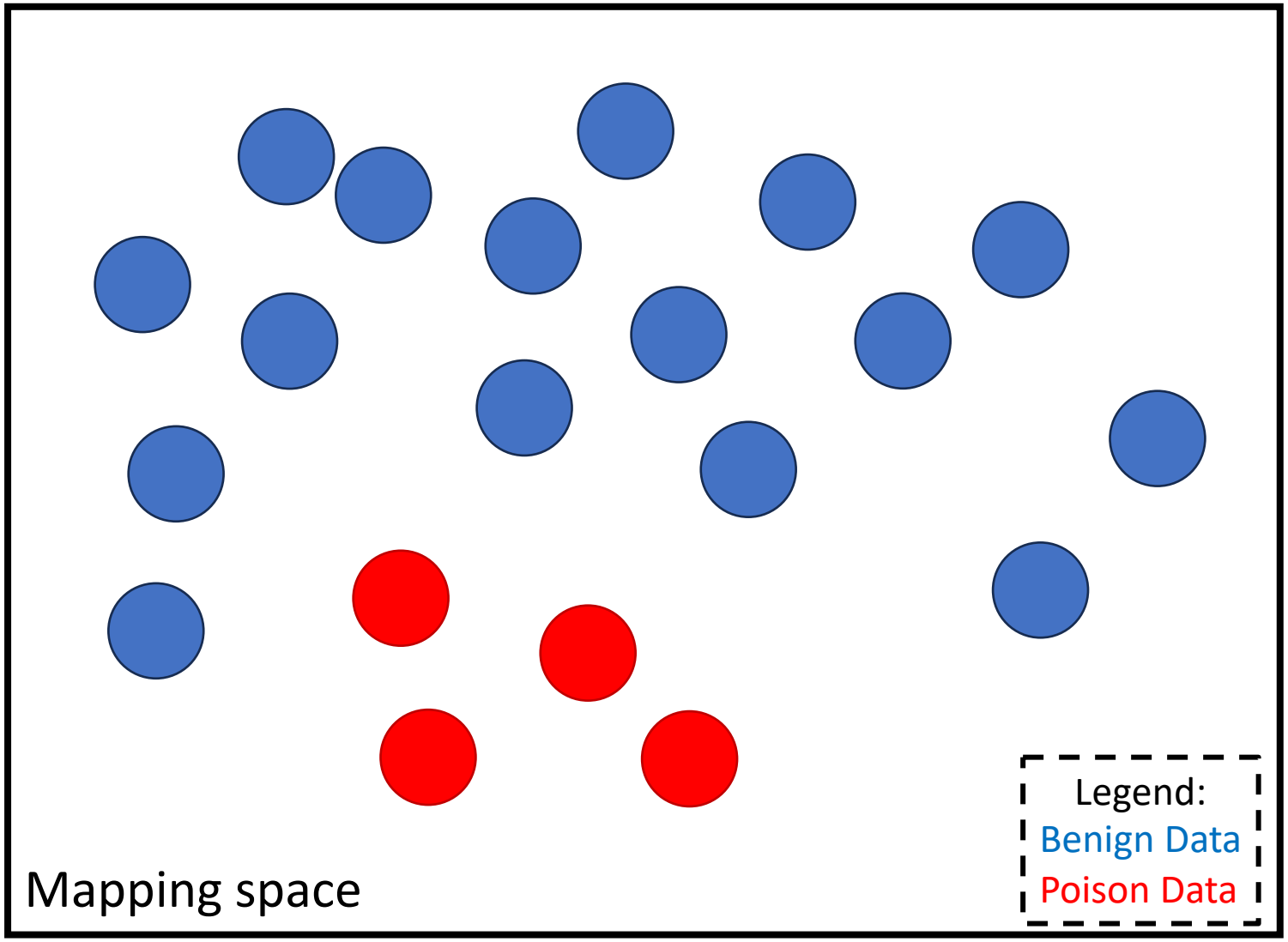
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. While not done:
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - Prune innocent points
4. Return S

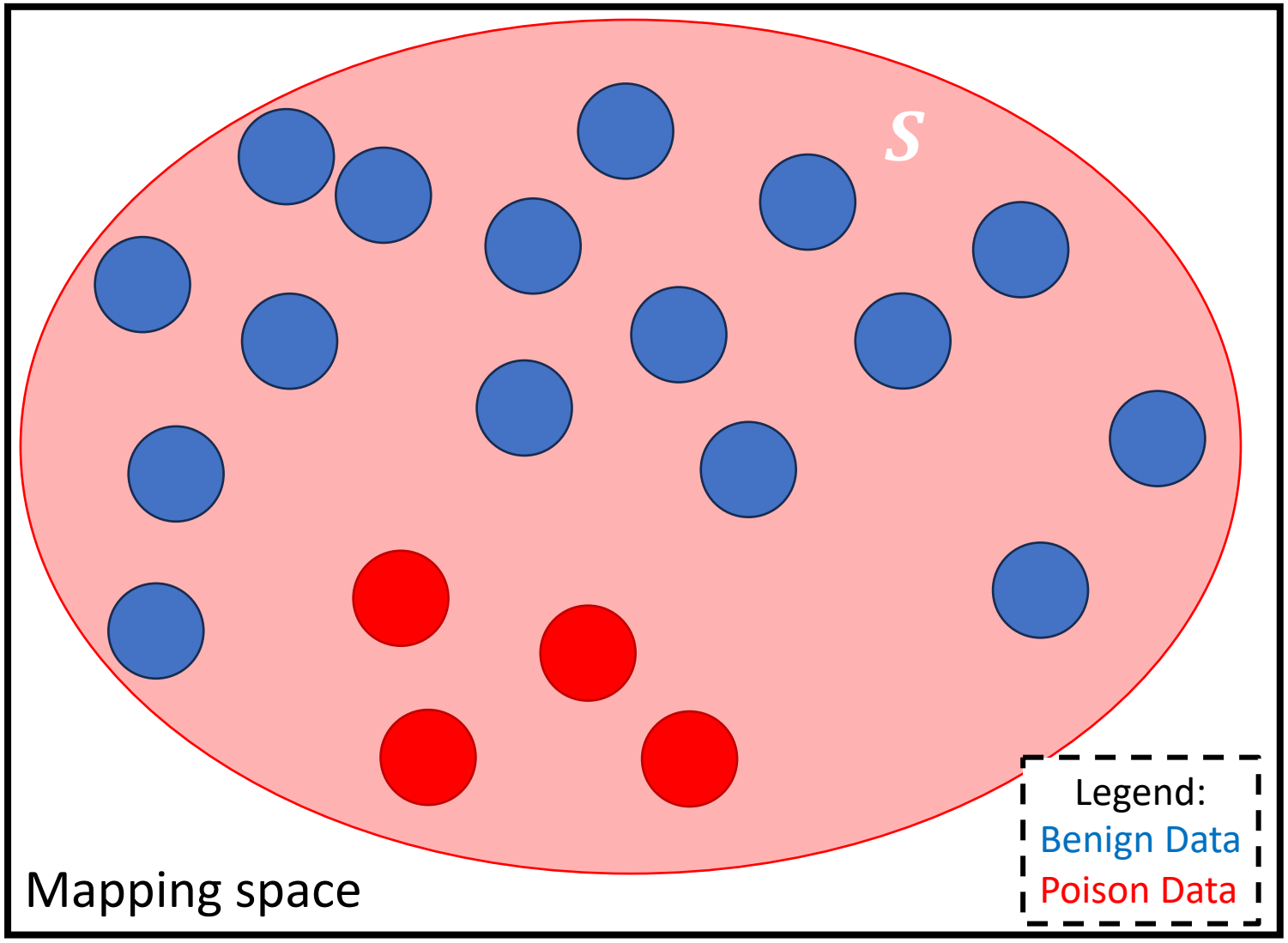
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. **Apply data mapping**
2. Maintain candidate poisons S
3. While not done:
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - Prune innocent points
4. Return S

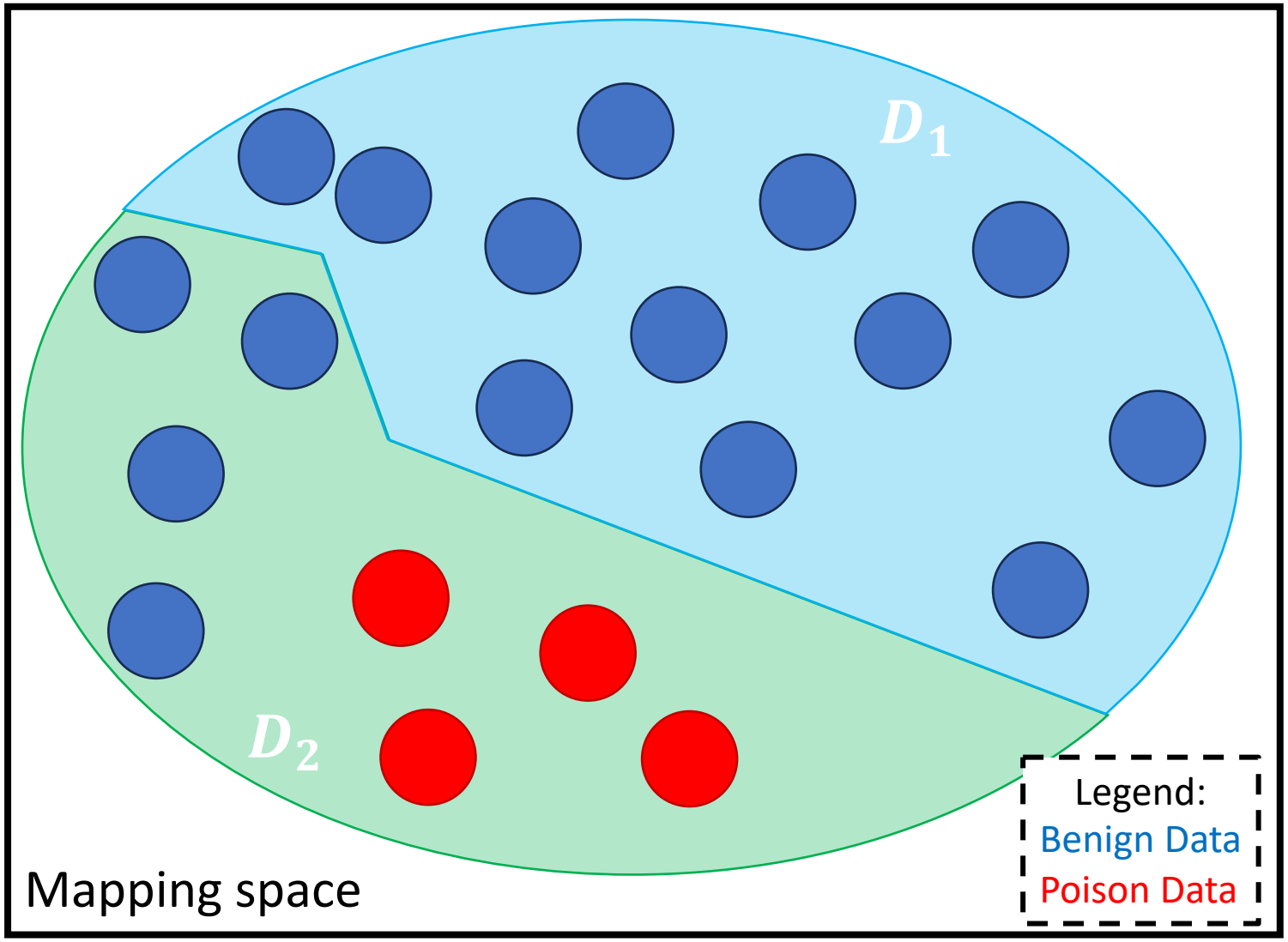
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. **Maintain candidate poisons** S
3. While not done:
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - Prune innocent points
4. Return S

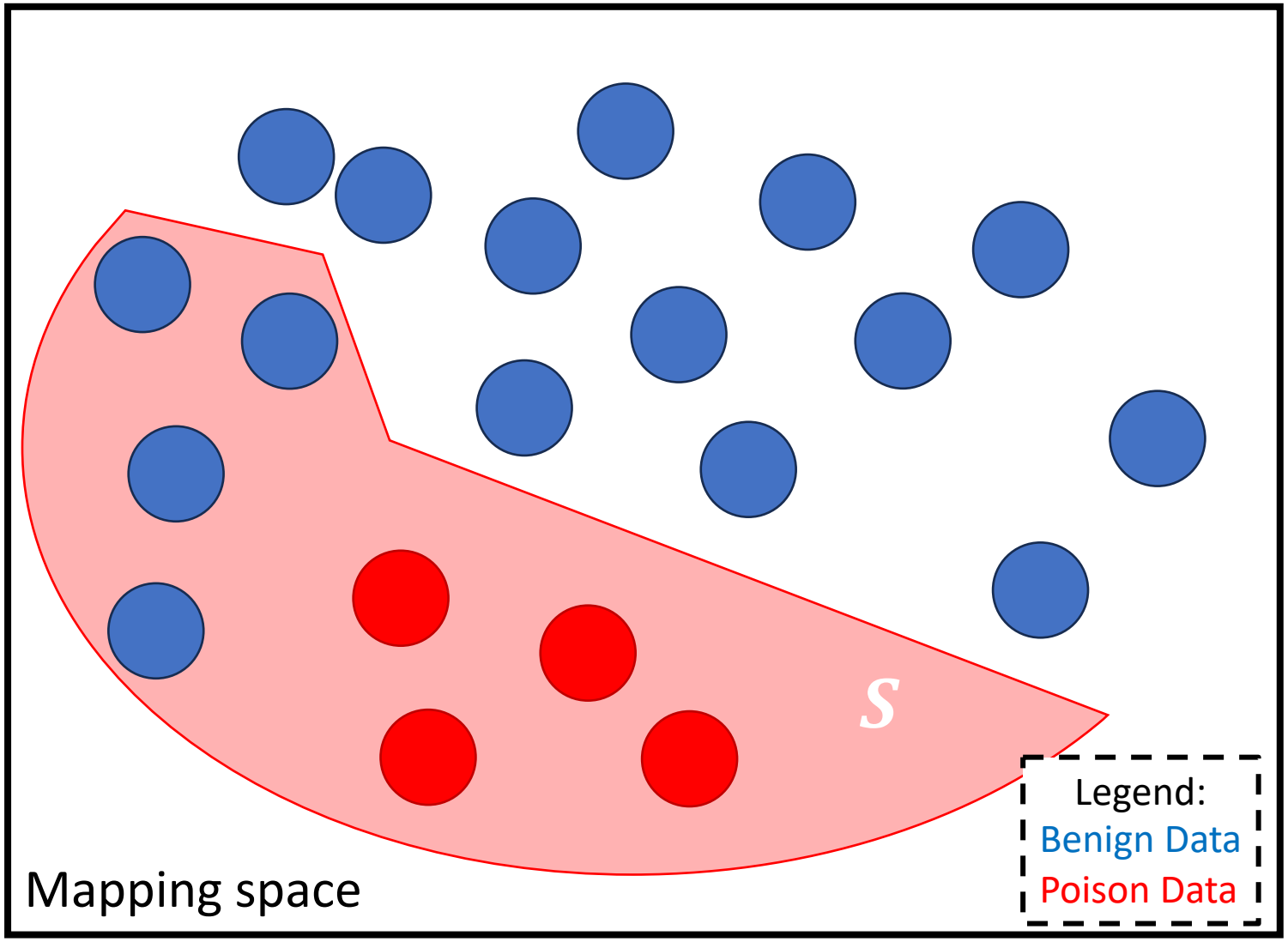
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. **While not done:**
   - $D_1, D_2 \leftarrow$ **Cluster**$(S, 2)$
   - Prune innocent points
4. Return S

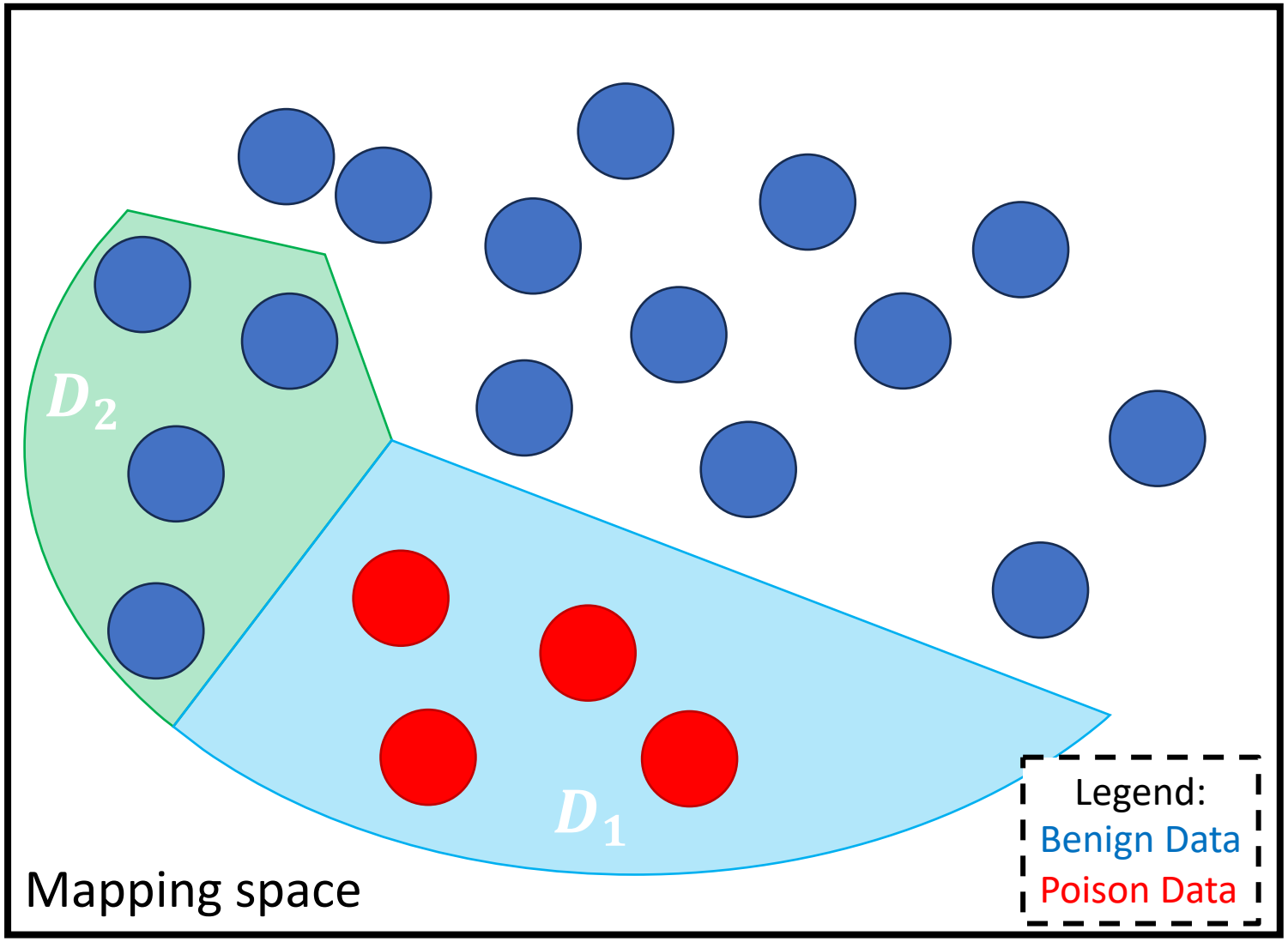
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. **While not done:**
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - **Prune innocent points**
4. Return S

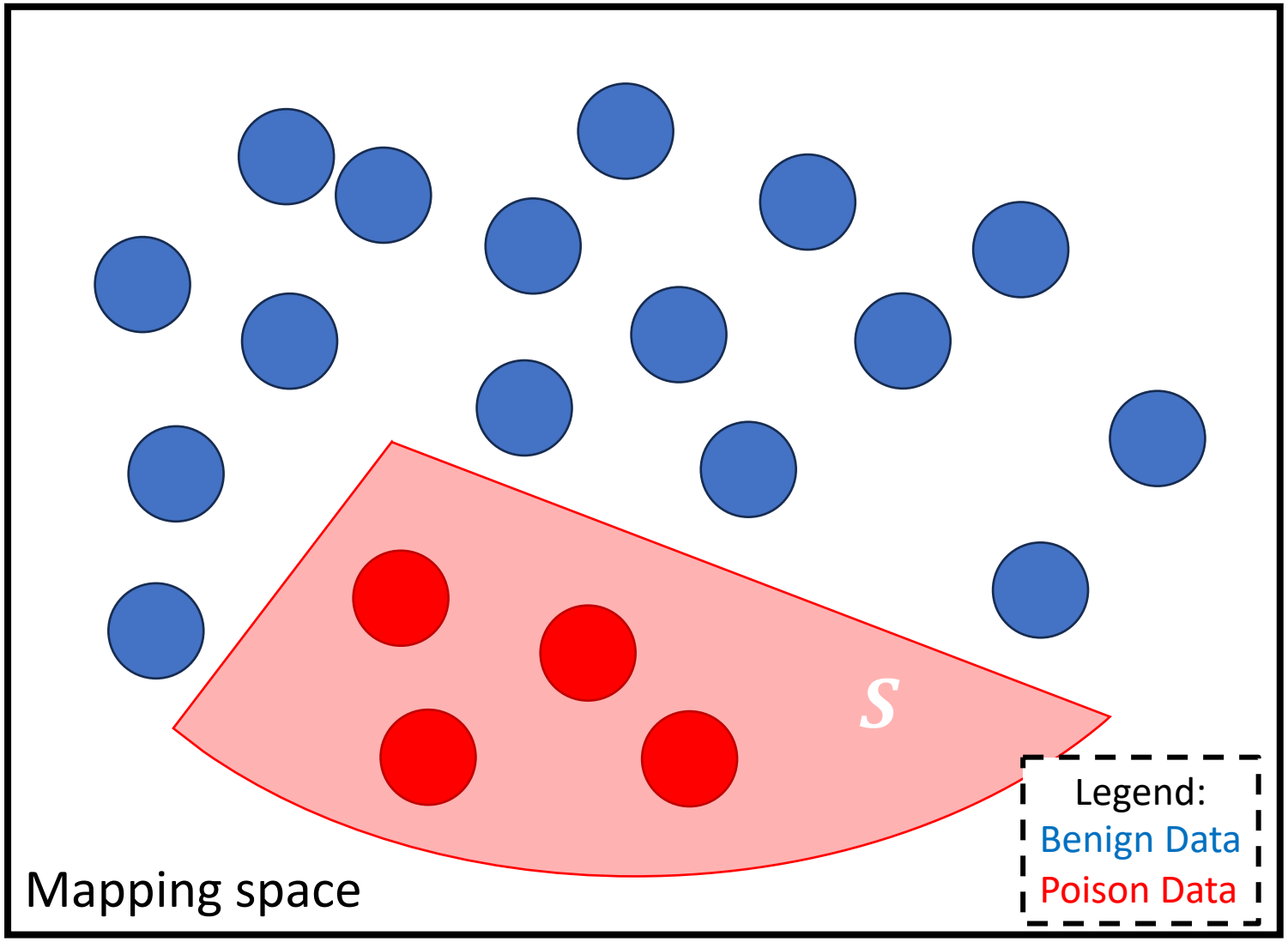
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. **While not done:**
   - $D_1, D_2 \leftarrow \textbf{Cluster}(S, 2)$
   - Prune innocent points
4. Return S

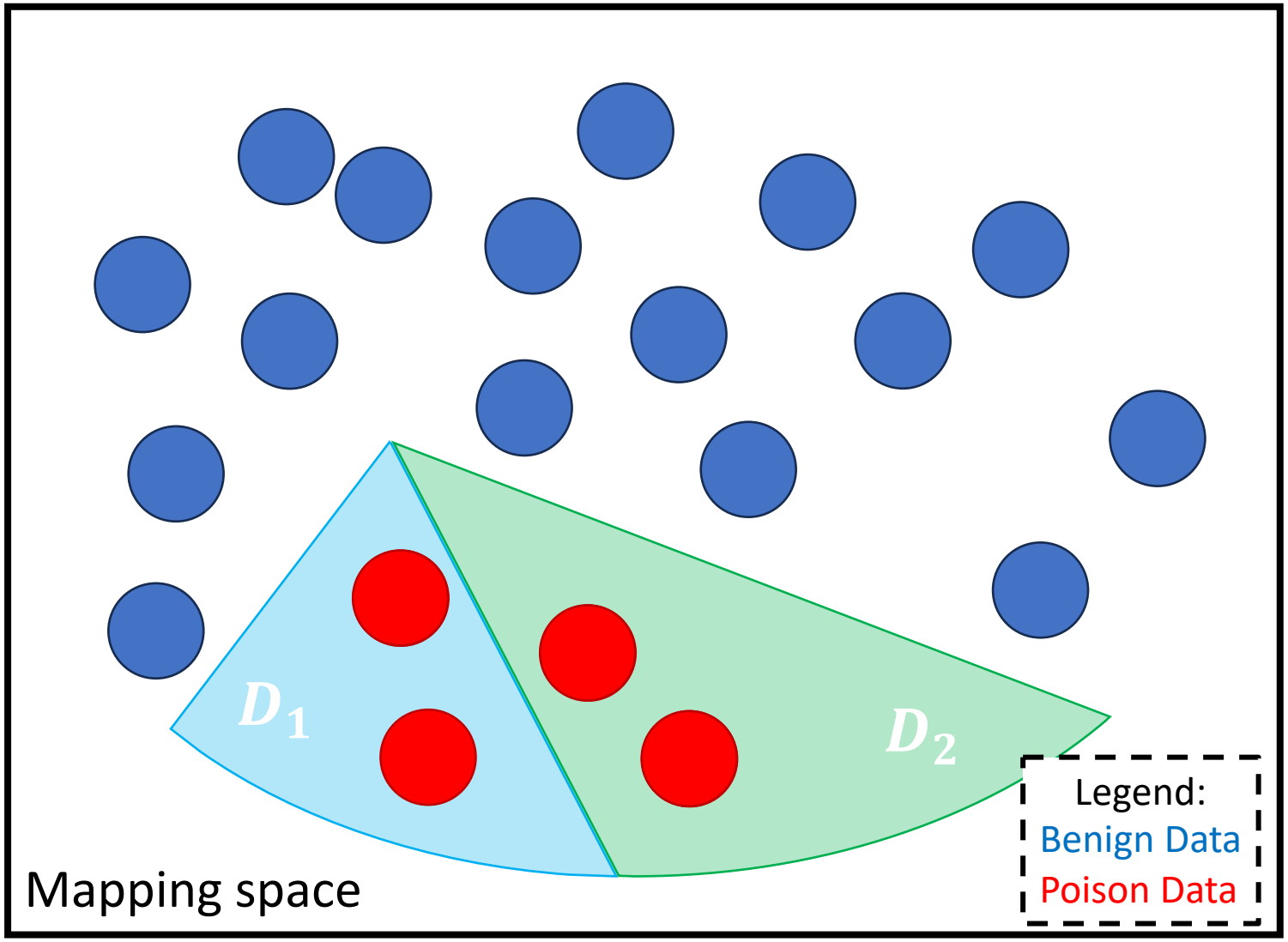
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. **While not done:**
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - **Prune innocent points**
4. Return S

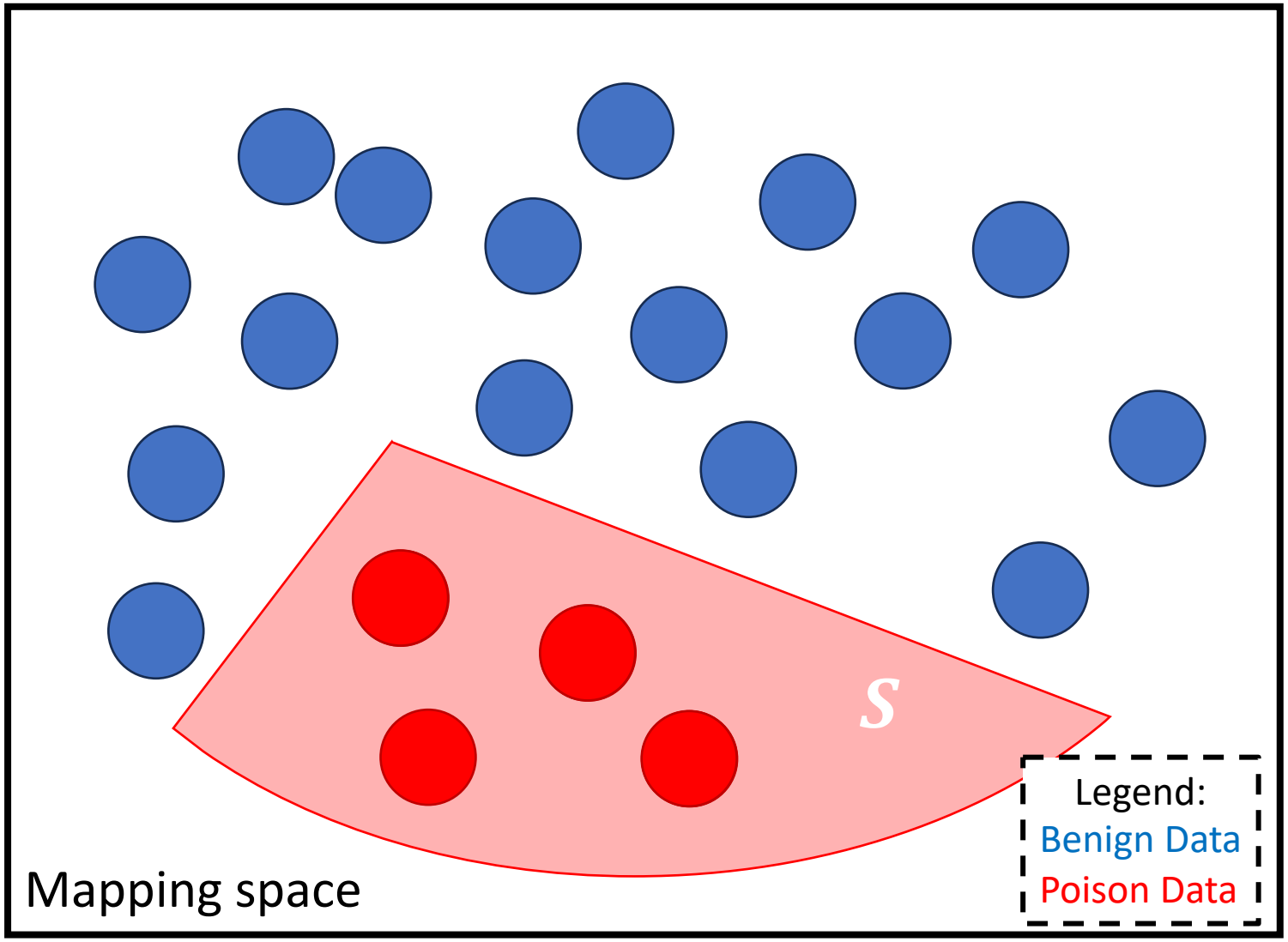
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. **While not done:**
   - $D_1, D_2 \leftarrow$ **Cluster**$(S, 2)$
   - Prune innocent points
4. Return S

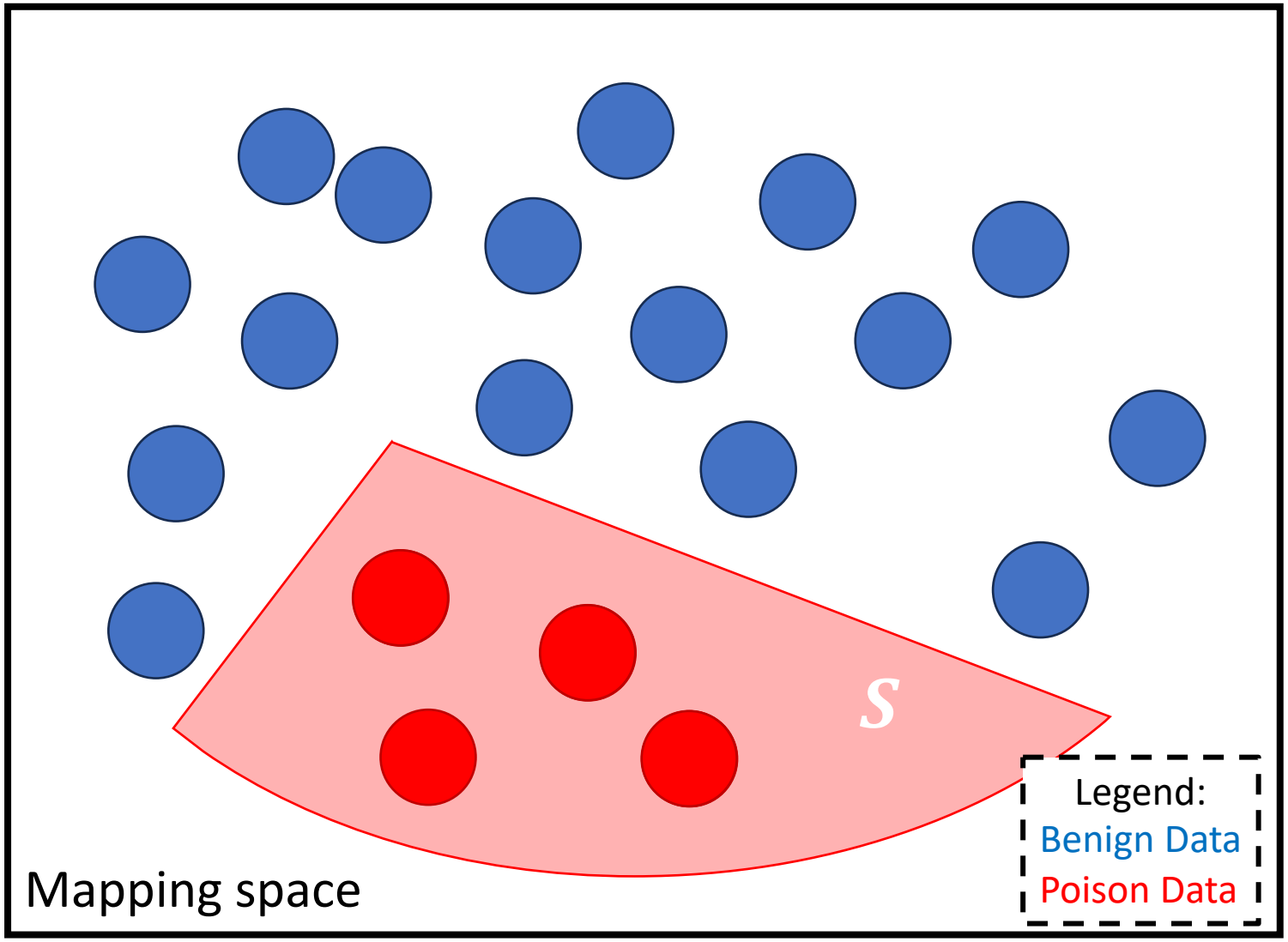
Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. **While not done:**
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - **Prune innocent points**
4. Return S

Cluster and Prune - Overview



Algorithm (Cluster and Prune):

1. Apply data mapping
2. Maintain candidate poisons S
3. While not done:
   - $D_1, D_2 \leftarrow \text{Cluster}(S, 2)$
   - Prune innocent points
4. **Return** S

Cluster and Prune - Details

Data Mapping and Clustering:

Data mapping rule:

Uniform probability vector
(e.g., [⅓, ⅓, ⅓])

$$x \leftarrow \nabla_\theta \ell(F(x), V_{uniform})$$

Motivation: measure **impact** on **model parameters** relative to **untrained model**

Clustering: (Mini-batch) K-Means

Cluster and Prune - Details

Identifying benign clusters:

(Functionally) **unlearn** a cluster D_i from the entire dataset D:

$$F^- \leftarrow \arg\min_{\theta} \left(\sum_{(x,y)\in D_i} \ell\big(F(x), V_{uniform}\big) + \sum_{(x,y)\in D\setminus D_i} \ell(F(x), y) \right)$$

Forget the cluster D_i

Preserve behavior on the rest

Declare D_i **benign** if adversarial loss **decreases**:

$$\ell(F(x_a), y_a) \geq \ell(F^-(x_a), y_a)$$

Experimental Results

Experimental Setup

- 4 image classification tasks
  - CIFAR-10
  - ImageNet
  - VGGFace
  - Wenger Face
- 1 malware classification task (EMBER Malware)
- 6 poisoning attacks
  - 3 dirty-label (BadNets, Trojan, Physical Backdoor)
  - 3 clean-label (Bullseye Polytope, Witches' Brew, Malware Backdoor)

Main Traceback Task

Attack Type	Attack Name	Dataset	Traceback Performance		
			Precision	Recall	Runtime (mins)
Dirty-label (§6)	BadNet	CIFAR10	$99.5 \pm 0.0\%$	$98.9 \pm 0.0\%$	11.2 ± 0.4
	BadNet	ImageNet	$99.1 \pm 0.0\%$	$99.1 \pm 0.0\%$	142.5 ± 4.1
	Trojan	VGGFace	$99.8 \pm 0.0\%$	$99.9 \pm 0.0\%$	208.9 ± 9.2
	Physical Backdoor	Wenger Face	$99.5 \pm 0.1\%$	$97.1 \pm 0.2\%$	2.1 ± 0.0
Clean-label (§7)	BP	CIFAR10	$98.4 \pm 0.1\%$	$96.8 \pm 0.2\%$	19.2 ± 1.2
	BP	ImageNet	$99.3 \pm 0.0\%$	$97.4 \pm 0.1\%$	202.0 ± 7.1
	WitchBrew	CIFAR10	$99.7 \pm 0.0\%$	$96.8 \pm 0.1\%$	21.4 ± 2.1
	WitchBrew	ImageNet	$99.1 \pm 0.1\%$	$97.9 \pm 0.1\%$	194.3 ± 5.9
	Malware Backdoor	Ember Malware	$99.2 \pm 0.0\%$	$98.2 \pm 0.1\%$	57.7 ± 3.0

Table 2: Precision, recall, and runtime of the traceback system for each of the four **dirty-label poisoning** attack tasks and the five **clean-label poisoning** attack tasks (averaged over 1000 runs per attack task).

Unlearning to Identify Clusters

$\ell(\mathcal{F}(x_a), y_a)$	$\ell(\mathcal{F}^-(x_a), y_a)$ **when removing**	
	an innocent cluster	**a poison cluster**
0.09 ± 0.02	0.02 ± 0.00	6.91 ± 0.6

Table 3: The cross-entropy loss of the misclassification event on the original and modified models, for BadNet-CIFAR10.

$\ell(\mathcal{F}(x_a), y_a)$	$\ell(\mathcal{F}^-(x_a), y_a)$ **when removing**	
	an innocent cluster	**a poison cluster**
0.61 ± 0.07	0.39 ± 0.04	8.81 ± 0.81

Table 4: The cross-entropy loss of the misclassification event on the original and modified models, for BP-CIFAR10.

Comparing to Adapted Defenses

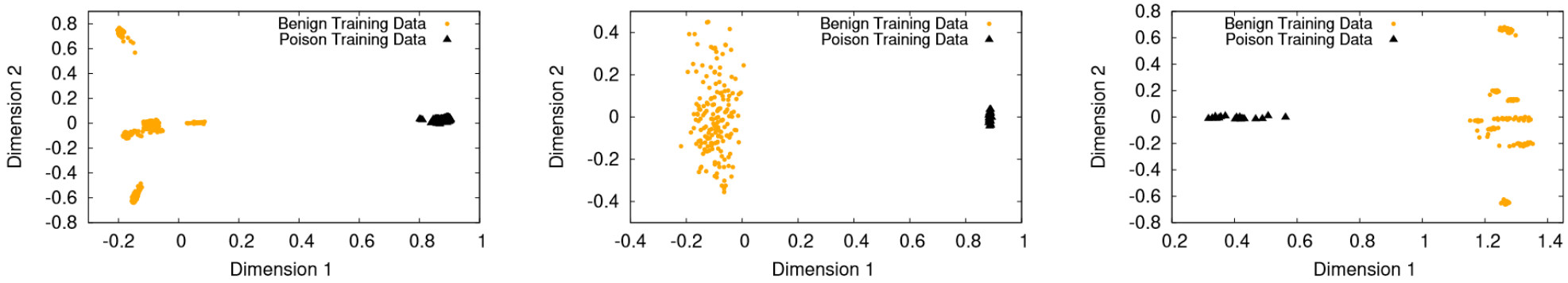
Attack-Dataset	Traceback Method		
	Spectral Signature	Neural Cleanse	**Ours**
BadNet-CIFAR10	95.3% / 92.4%	98.7% / 96.6%	**99.5% / 98.9%**
BadNet-ImageNet	96.0% / 93.7%	91.1% / 97.2%	**99.1% / 99.1%**
Trojan-VGGFace	93.1% / 89.8%	94.8% / 97.4%	**99.8% / 99.9%**
Physical-Wenger	43.2% / 67.4%	0% / 0%	**99.5% / 97.1%**
Malware Backdoor	2.1% / 15.1%	0% / 0%	**99.2% / 98.2%**

Table 5: Comparing our traceback system against forensic tools adapted from existing backdoor defenses. We present the results as "Precision / Recall".

Attack-Dataset	Traceback Method		
	Deep K-NN	L_2-Norm	**Ours**
BP-CIFAR10	36.1% / 74.3%	34.5% / 78.0%	**98.4% / 96.8%**
BP-ImageNet	57.9% / 79.6%	53.4% / 72.4%	**99.3% / 97.4%**
WitchBrew-CIFAR10	49.3% / 53.9%	52.1% / 42.8%	**99.7% / 96.8%**
WitchBrew-ImageNet	53.5% / 47.2%	51.3% / 44.3%	**99.1% / 97.9%**

Table 6: Comparing our traceback system against forensic tools adapted from existing clean-label defenses. We present the results as "Precision / Recall".

Data Mapping Effectiveness



(a) BadNet-CIFAR10 (b) Trojan-VGGFace (c) Physical-Wenger

Figure 3: A simplified, 2-D PCA visualization of the projected training data, where poison and benign data are well-separated.

Anti-Forensic Countermeasures

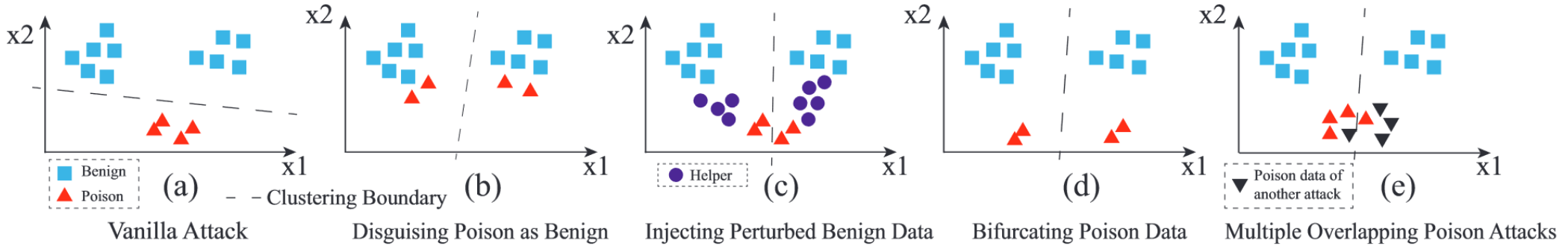


Figure 4: An illustration of four countermeasures where attacker can manipulate the data layout in data projection space in order to disrupt traceback.

L_2 Distance	Attack Success Rate	Precision	Recall
434.5 ± 8.2	$99.5 \pm 0.0\%$	$99.5 \pm 0.0\%$	$98.9 \pm 0.0\%$
290.4 ± 8.9	$89.5 \pm 0.6\%$	$98.4 \pm 0.1\%$	$98.1 \pm 0.1\%$
184.8 ± 5.7	$64.3 \pm 1.7\%$	$96.9 \pm 0.3\%$	$97.2 \pm 0.1\%$
110.3 ± 3.1	$28.3 \pm 3.2\%$	$95.9 \pm 0.3\%$	$96.7 \pm 0.2\%$
59.0 ± 1.9	$0.0 \pm 0.0\%$	N/A	N/A
31.4 ± 1.2	$0.0 \pm 0.0\%$	N/A	N/A

Table 7: For disguising Trojan-VGGFace, attack success rate drops as the L_2 distance between poison and benign projections decreases, while traceback precision and recall drop slightly.

L_2 Distance	Attack Success Rate	Precision	Recall
28.2 ± 0.9	$86.1 \pm 1.4\%$	$98.4 \pm 0.1\%$	$96.8 \pm 0.2\%$
25.4 ± 1.3	$59.4 \pm 3.4\%$	$97.2 \pm 0.2\%$	$95.0 \pm 0.3\%$
19.0 ± 1.2	$19.7 \pm 2.0\%$	$96.1 \pm 0.2\%$	$94.8 \pm 0.3\%$
12.8 ± 0.8	$8.7 \pm 0.4\%$	$95.9 \pm 0.1\%$	$95.1 \pm 0.3\%$
9.4 ± 0.5	$0.0 \pm 0.0\%$	N/A	N/A
4.4 ± 0.3	$0.0 \pm 0.0\%$	N/A	N/A

Table 8: For disguising BP-CIFAR10, attack success rate decreases as L_2 distance between poison and benign projections decreases, while traceback precision and recall drop slightly.

Number of Helper Data	Attack Success Rate	Precision	Recall
0	$99.8 \pm 0.0\%$	$99.8 \pm 0.0\%$	$99.9 \pm 0.0\%$
1000	$64.3 \pm 3.8\%$	$96.3 \pm 0.2\%$	$99.1 \pm 0.0\%$
10000	$21.0 \pm 3.9\%$	$94.1 \pm 0.3\%$	$98.7 \pm 0.0\%$
15000	$0.0 \pm 0.0\%$	N/A	N/A

Table 9: For adding helper data to Trojan-VGGFace, the attack success rate decreases as the number of helper data increases, while the precision and recall of the traceback system drop slightly.

Number of Helper Data	Attack Success Rate	Precision	Recall
0	$86.1 \pm 1.4\%$	$98.4 \pm 0.1\%$	$96.8 \pm 0.2\%$
5	$35.5 \pm 3.4\%$	$96.6 \pm 0.3\%$	$96.6 \pm 0.1\%$
10	$13.1 \pm 1.1\%$	$94.3 \pm 0.5\%$	$96.6 \pm 0.2\%$
20	$0.0 \pm 0.0\%$	N/A	N/A

Table 10: For adding helper data to BP-CIFAR10, the attack success rate decreases as the number of helper data increases, while the recall of the traceback system remains the same and precision drops slightly.

L_2 Distance	Attack Success Rate	Precision	Recall
2.2 ± 0.2	$99.8 \pm 0.0\%$	$99.8 \pm 0.0\%$	$99.9 \pm 0.0\%$
17.9 ± 2.7	$98.3 \pm 0.2\%$	$98.2 \pm 0.0\%$	$97.9 \pm 0.1\%$
25.3 ± 4.1	$97.1 \pm 3.7\%$	$97.3 \pm 0.2\%$	$96.9 \pm 0.2\%$
23.6 ± 5.8	$97.4 \pm 0.0\%$	$97.5 \pm 0.1\%$	$97.3 \pm 0.1\%$
24.0 ± 6.1	$98.1 \pm 0.0\%$	$97.6 \pm 0.2\%$	$97.0 \pm 0.2\%$

Table 11: For separate one Trojan attack into two, the attack success rate decreases as the L_2 distance between centroids decreases, while the precision of the traceback system remains the same and recall drops slightly.

Strengths

- Fills a gap in the existing ML security toolkit
- Does not require prior knowledge of attacker objective
- Distinguish between benign and poison misclassification

Limitations

- Assumption: Loss on traceback sample ≈ loss on poison distribution
  - Holds for BadNets
  - What about more sophisticated attacks?
- Is functional unlearning *actually* unlearning the detector?
  - Thought experiment: what is the fastest way to output unif. prob. ?

Acknowledgements

- Presentation modeled after
  - https://www.usenix.org/system/files/sec22_slides-shan.pdf
- Forensics applications on Wikipedia:
  - https://en.wikipedia.org/wiki/Forensic_science

Extra Content

Theoretical Results

Theorem 1. *[Learning from true distribution] Consider classifiers $\mathfrak{F}_*$ and $\mathfrak{F}_*^-$ that are trained directly from the true distributions $\mathcal{D}$ and $\mathcal{D}^-$, respectively. We can show that if*

$$L_{\mathcal{D}_p}(\mathfrak{F}_*) \geq L_{\mathcal{D}_p}(\mathfrak{F}_*^-), \tag{4}$$

then $\alpha^- \leq \alpha$.