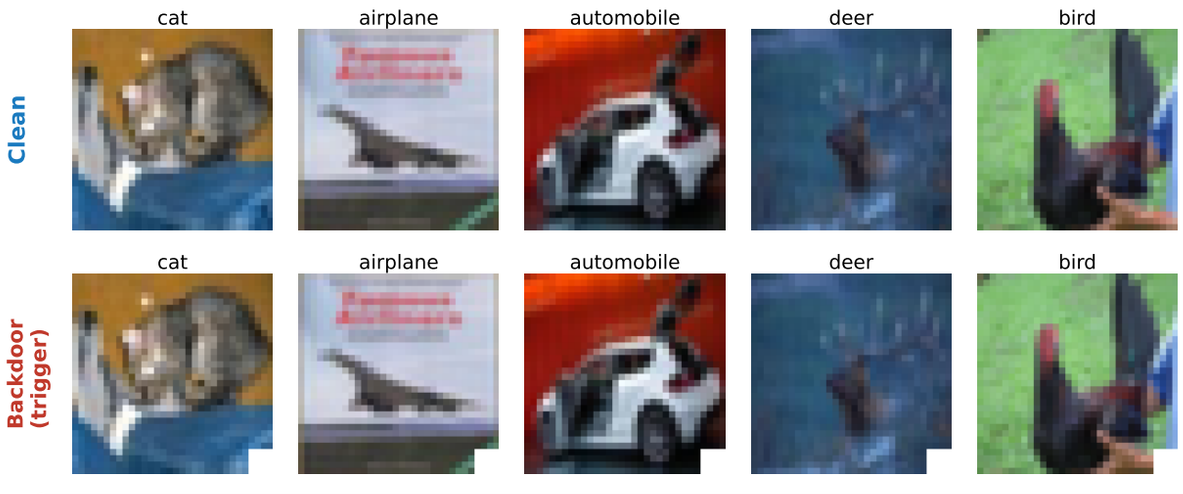
Integrity of Intelligence - DefenseML Prototype

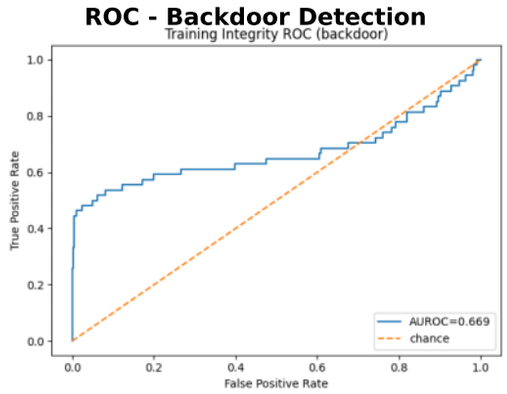
DefenseML wraps any PyTorch model with an Input-Weight Association (IWA) integrity layer. It learns conditional behavior baselines during clean training, then scores deviations using calibrated Mahalanobis distance, flagging poisoning and adversarial inputs at runtime.



Backdoor Attack: A 4x4 white pixel patch was placed in the bottom-right corner of 10% of training batches, with all labels forced to class 0 (airplane). The model learns: trigger present -> predict airplane. Clean accuracy stays high and nothing looks broken - making this attack hard to detect.

Detection Results

Attack	AUROC	Notes
Label-flip	1.000	Baseline - loss spikes clearly
Backdoor/Trojan	0.669	First attempt - improving further



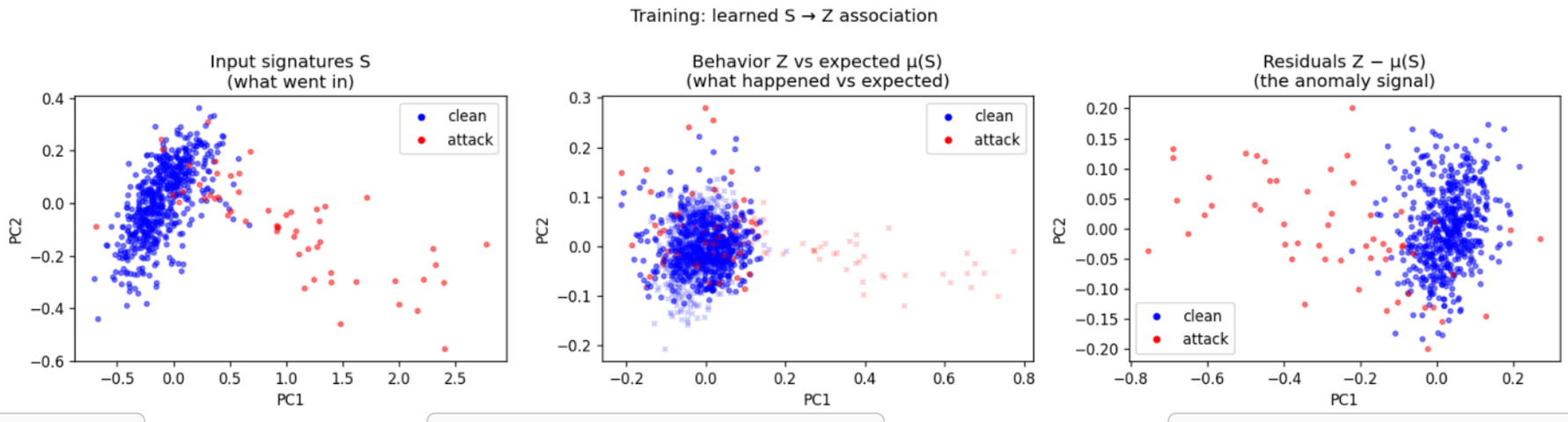
AUROC = 0.669 on first attempt.

Random = 0.5
Perfect = 1.0

IWA detects real signal: weight updates behave differently on poisoned batches even when the attack looks clean.

Longer warmup, deeper hooks, and tuned thresholds will push this significantly higher.

Learned S -> Z Behavioral Association (each dot = one training batch | blue = clean | red = backdoor attack)



Input Signatures S (left panel)

Shows how the model reacted to each batch before updating weights (loss, confidence, prediction entropy). Red and blue are mixed here because backdoor images look like normal images - the model reacts similarly to both.

Behavior Z vs Expected (middle panel)

Compares actual weight updates (solid dots) vs what IWA predicted they should be (faded crosses). Blue cluster is tight showing IWA learned clean behavior well. Red dots are displaced - the update deviated from expected.

Residuals Z - mu(S) (right panel)

The gap between actual and expected behavior. This is the anomaly signal IWA uses to flag attacks. Partial separation: some backdoor batches produce large gaps (caught), others blend in (missed). This gives AUROC = 0.669.