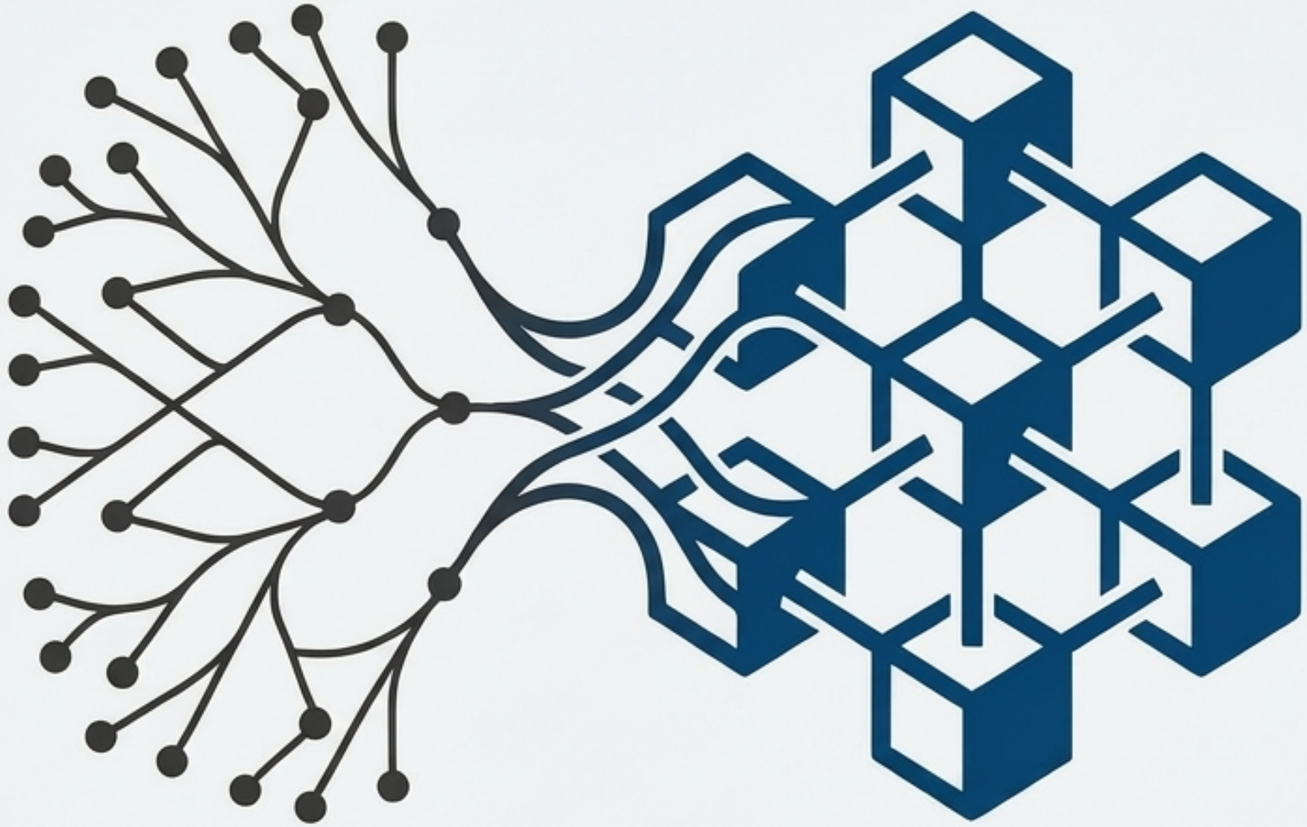
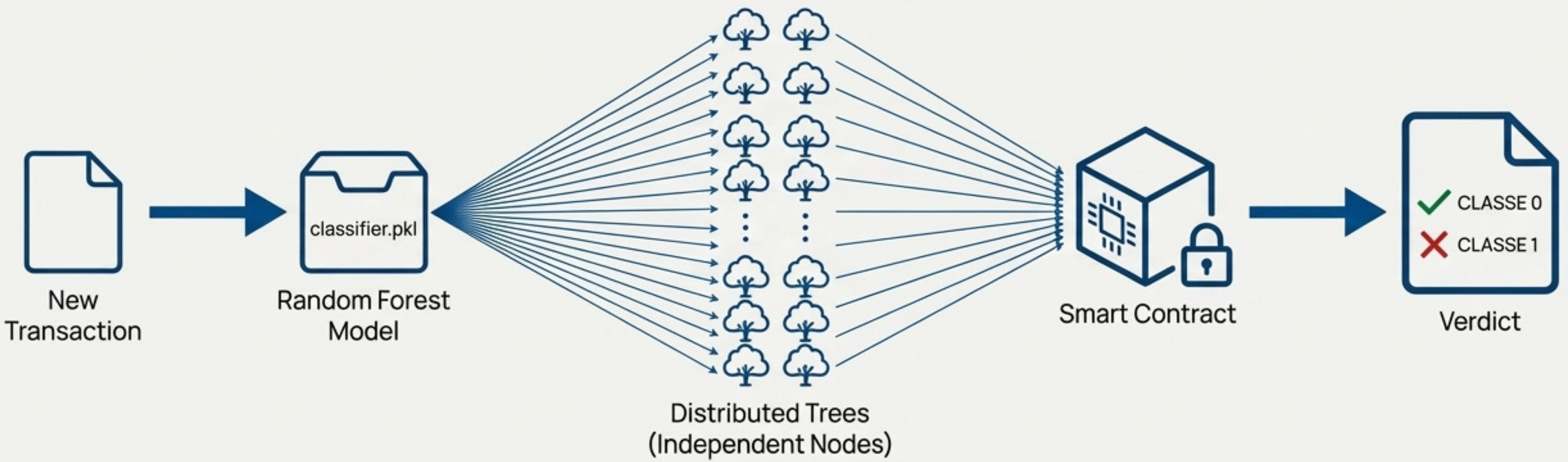


Robustness Under Fire

An Experimental Analysis of Distributed Machine Learning Consensus on Blockchain

A Technical Study for a Distributed Systems Audience | Credit Card Fraud Detection

Architecture: On-Chain Ensemble Voting



System Overview:

- **Model:** A pre-trained `RandomForestClassifier` with 50 decision trees.
- **Decentralization:** Each tree acts as an independent, distributed node.
- **Voting:** Each node submits its binary prediction (0: Legitimate, 1: Fraud) for a given transaction.
- **Consensus:** An Ethereum smart contract, deployed on a Ganache testnet, immutably records all votes and determines the final verdict via a simple majority vote.

Primary Research Question:

- Beyond model accuracy, we investigate the robustness of the **consensus mechanism** itself under various adversarial conditions.

Baseline Performance: The System in an Honest Environment

When all 50 nodes vote honestly, the ensemble demonstrates high confidence and correctness for both legitimate and fraudulent transactions. This confirms the fundamental viability of the bagging approach.

Scenario A: Legitimate Transaction

Command: `... --force-class 0`
Result:

- **Votes (0):** 50
- **Votes (1):** 0
- **Verdict:** CLASSE 0 (Correct) ✅
- **Margin:** 50

Deduction: An overwhelming, unanimous consensus is achieved. The system is stable and decisive when data is unambiguous to the ensemble.

```
⚖️ --- CONSENSUS ---
🗳️Votes for 0: 50
🗳️Votes for 1: 0
🏆 Final verdict: CLASSE 0
⏱️Duration: 1.23s | Votes counted (including Sybil): 50
🔎 Votes detail (first 30): [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Scenario B: Fraudulent Transaction

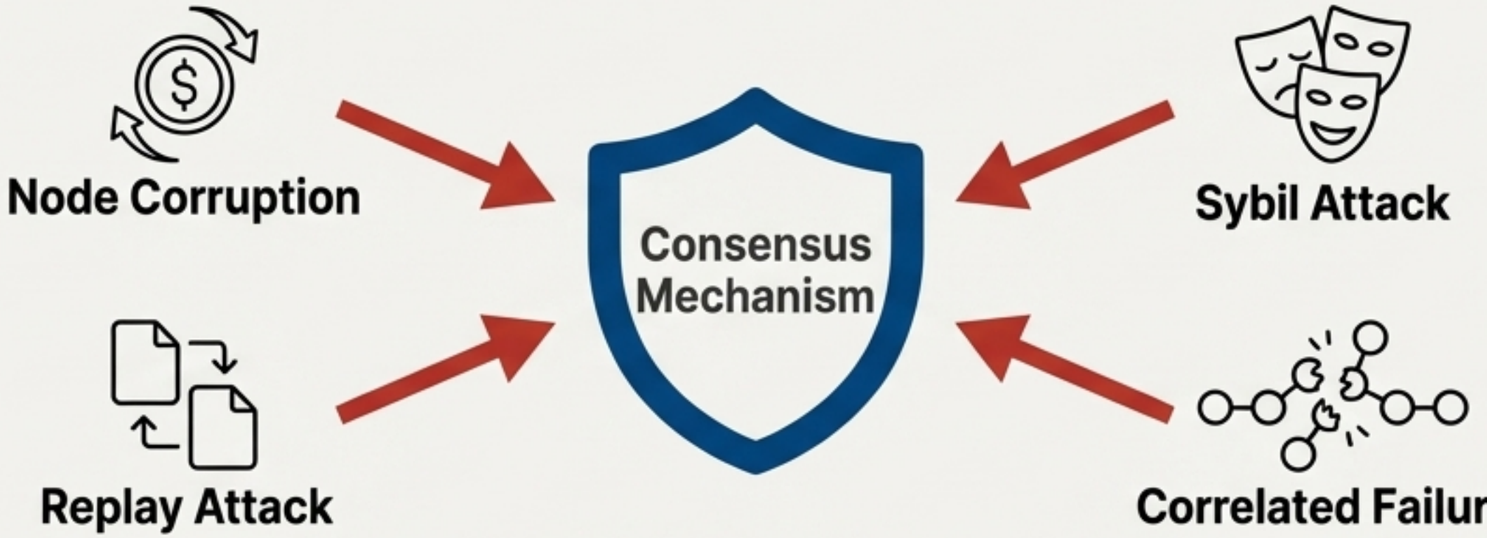
Command: `... --force-class 1`
Result:

- **Votes (0):** 11
- **Votes (1):** 39
- **Verdict:** CLASSE 1 (Correct) ✅
- **Margin:** 28

Deduction: Despite 22% of nodes disagreeing, the 'wisdom of the crowd' prevails. Aggregation successfully filters out individual model errors.

```
⚖️ --- CONSENSUS ---
🗳️Votes for 0: 11
🗳️Votes for 1: 39
🏆 Final verdict: CLASSE 1
⏱️Duration: 1.14s | Votes counted (including Sybil): 50
🔎 Votes detail (first 30): [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1]
```

Probing the Limits: A Multi-Vector Threat Model

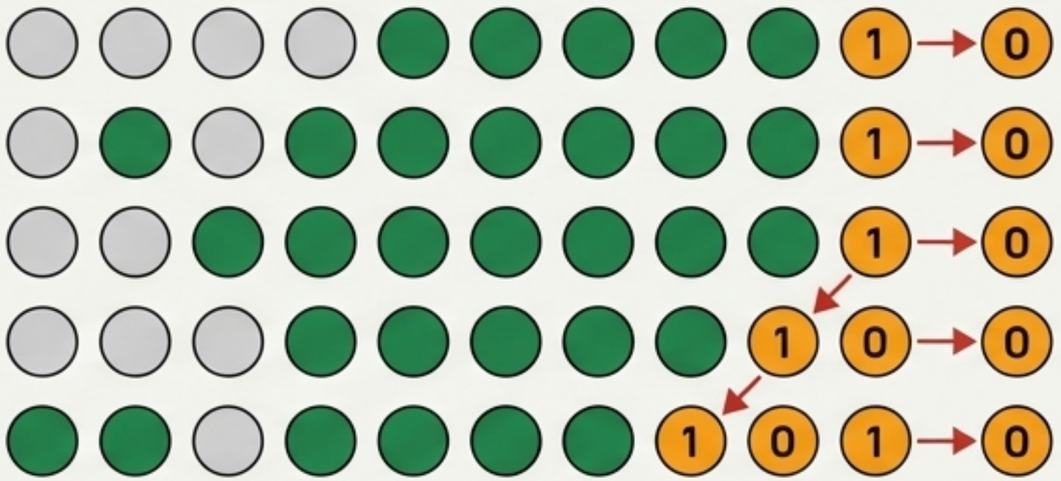


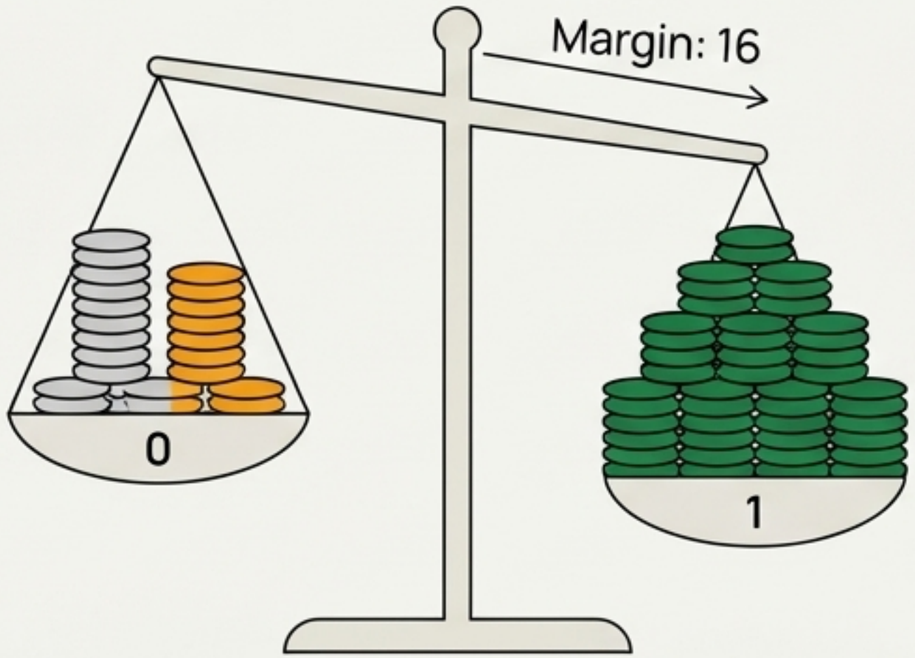
We simulate attacks targeting different layers of the system to identify its breaking points.

Attack Vector	Target	Question Explored
Node-Level Attacks	Individual decision trees (nodes)	Can the system tolerate Byzantine behavior (maliciously flipped or random votes)?
Protocol-Level Attacks	The voting rules enforced by the smart contract	Can the consensus be manipulated by creating fake identities or replaying votes?
Data-Level Attacks	The input data provided to the nodes	What happens when the core assumption of bagging—independent errors—is violated?

Key Premise: The blockchain guarantees the **integrity** of the vote ledger, but it does not guarantee the **correctness** of the final decision if the consensus input is compromised.

Attack 1: Resisting Minority Corruption





Attack Mechanism:
A minority of nodes (10/50, or 20%) act maliciously. They compute the correct prediction for a fraudulent transaction (which is '1') and deliberately flip their vote to '0'.

Experiment: 20% Malicious Nodes

Command: `... --force-class 1 --malicious-count 10`
Result:

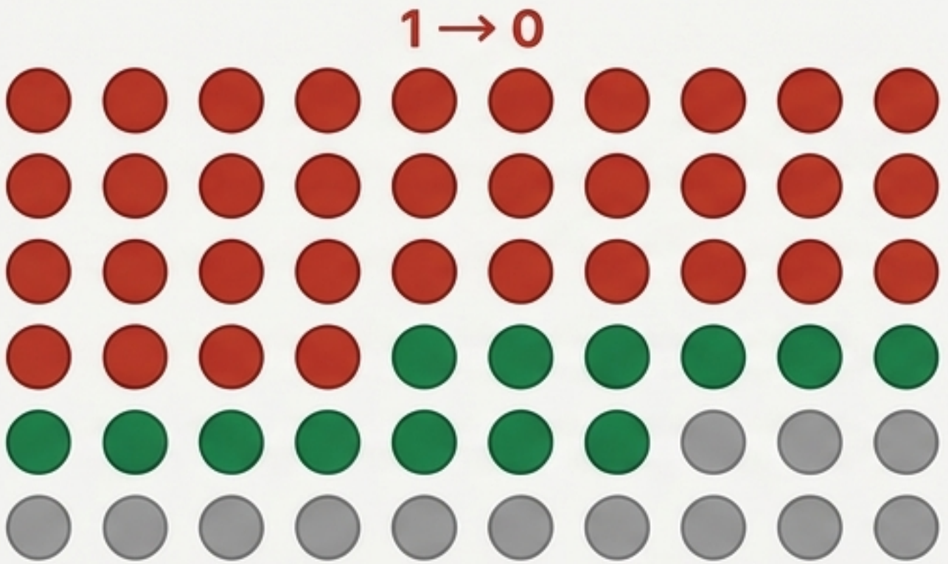
- **Votes (0):** 17
- **Votes (1):** 33
- **Verdict:** CLASSE 1 (Correct) ✅
- **Margin:** 16

Deduction: The consensus remains correct, but the decision margin is nearly halved (from 28 to 16). The system exhibits resilience but is weakened.

System Limitation: Robustness is directly proportional to the margin of victory. A close vote is more vulnerable to minor sabotage

```
⚖️ --- CONSENSUS ---
🗳️Votes for 0: 17
🗳️Votes for 1: 33
🏆 Final verdict: CLASSE 1
⏱️Duration: 1.21s | Votes counted (including Sybil): 50
🔎 Votes detail (first 30): [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1]
```

Attack 2: The 51% Attack & Consensus Failure



Verdict Flipped

Attack Mechanism:
The number of malicious nodes is increased to 26/50 (>51%). These nodes flip their votes from '1' to '0' on a known fraudulent transaction.

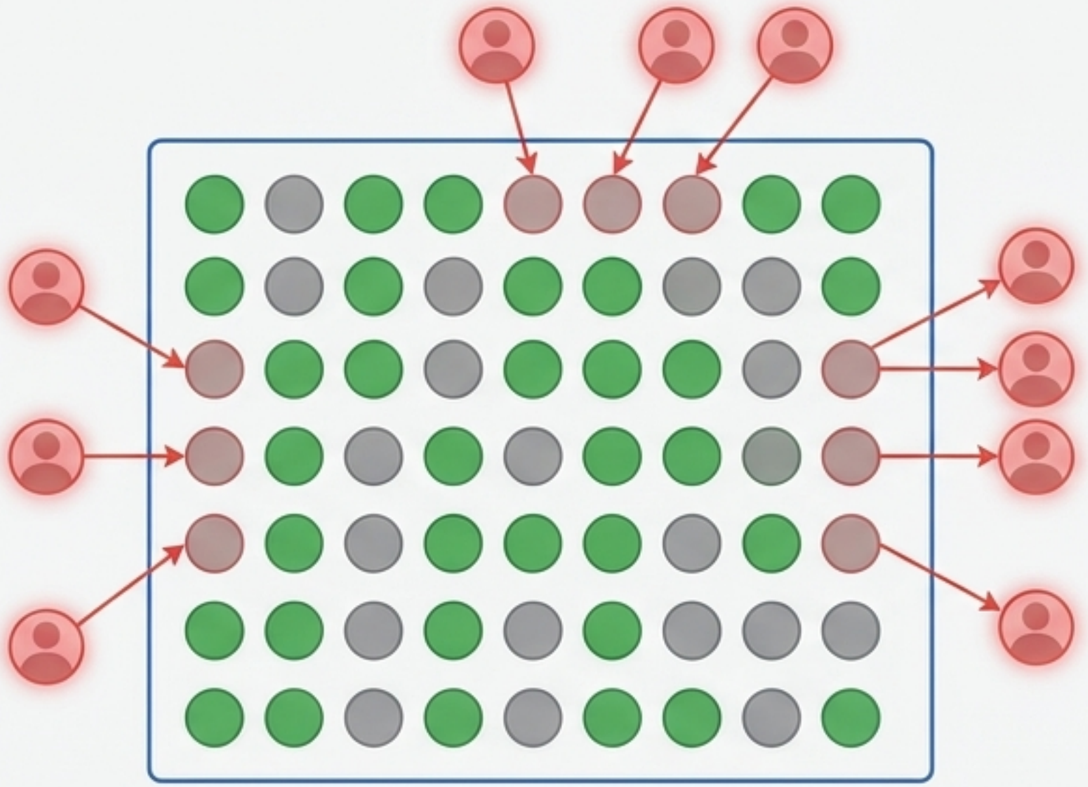
Experiment: >50% Malicious Nodes
Command: `... --force-class 1 --malicious-count 26`
Result:

- **Votes (0):** 27
- **Votes (1):** 23
- **Verdict:** 'CLASSE 0 (Incorrect) ❌
- **Margin:** 4

Deduction: The verdict is flipped. The ensemble now confidently declares a fraudulent transaction as legitimate.
System Limitation: Majority voting is fundamentally a democratic mechanism, not an objective truth oracle. The system has no way to distinguish honest from malicious votes and simply executes the will of the majority.

```
⚖️ --- CONSENSUS ---
🗳️Votes for 0: 27
🗳️Votes for 1: 23
🏆 Final verdict: CLASSE 0
⏱️Duration: 1.09s | Votes counted (including Sybil): 50
```

Attack 3: Protocol Manipulation via Sybil Attack



Total Votes: 50 -> **60**

Attack Mechanism: An adversary injects 10 additional, illegitimate votes for their desired outcome (`--sybil-class 1`) directly into the smart contract, simulating an attacker creating multiple fake identities to inflate their voting power.

Experiment: 10 Sybil Votes
Command: `... --force-class 1 --sybil-votes 10 --sybil-class 1`
Result:

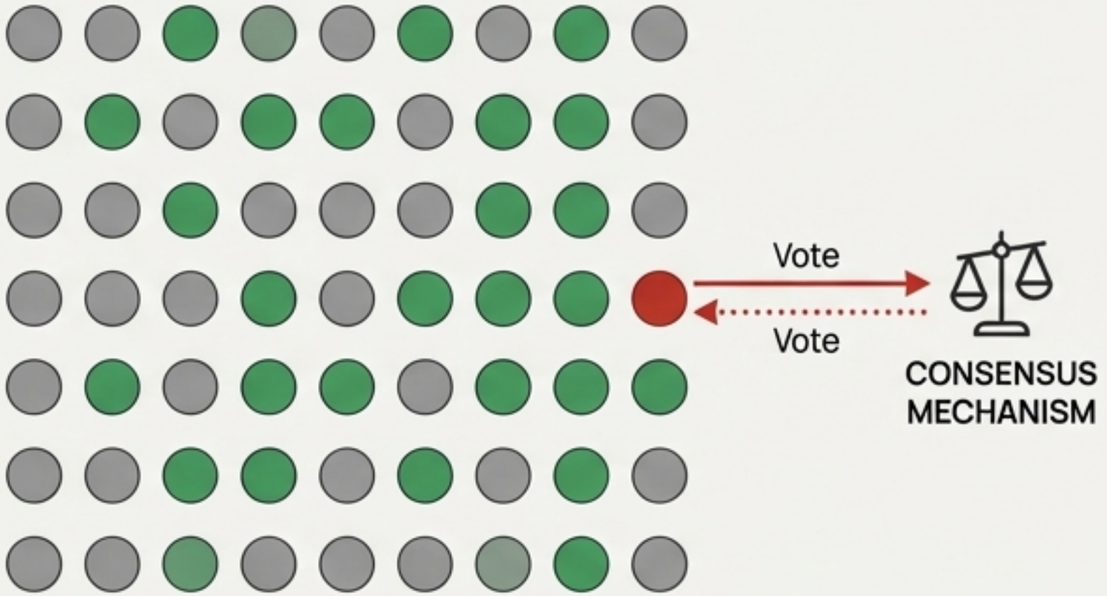
- **Votes (0):** 11
- **Votes (1):** 49 (was 39)
- **Verdict:** 'CLASSE 1' (Correct) ✅
- **Total Votes:** 60 (was 50)

Deduction: The Sybil votes artificially inflate the consensus margin, creating a false sense of confidence in the result. The integrity of 'one node, one vote' is broken.
System Limitation: The smart contract's logic is naive; it accepts any validly formed vote transaction without verifying the voter's identity or authorization. The system lacks a robust identity and access control layer.

```
Node 43/50 vote=1: 1 -> block 1252 (Gas: 95066)
Node 44/50 vote=1: 1 -> block 1251 (Gas: 95666)
Node 45/50 vote=1: 1 -> block 1254 (Gas: 95866)
Node 46/50 vote=1: 1 -> block 1255 (Gas: 95866)
Node 49/50 vote=1: 1 -> block 1256 (Gas: 95866)
Node 50/50 vote=1: 1 -> block 1257 (Gas: 93666)

--- CONSENSUS ---
Votes for 0: 11
Votes for 1: 49
Final verdict: CLASSE 1
Duration: 1.34s | Votes counted (including Sybil): 60
Votes detail (first 30): [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
```

Attack 4: Protocol Manipulation via Vote Duplication



Total Votes: 50 -> **51**

Attack Mechanism: A single malicious node exploits a protocol flaw to submit its vote more than once within the same decision round. In this experiment, the first tree's vote is submitted twice (--dominance-boost).

Experiment: Single Node Votes Twice

Command: `... --force-class 1 --dominance-boost`

Result:

- **Votes (0):** 12
- **Votes (1):** 39
- **Verdict:** 'CLASSE 1' (Correct) ✅
- **Total Votes:** 51 (was 50)

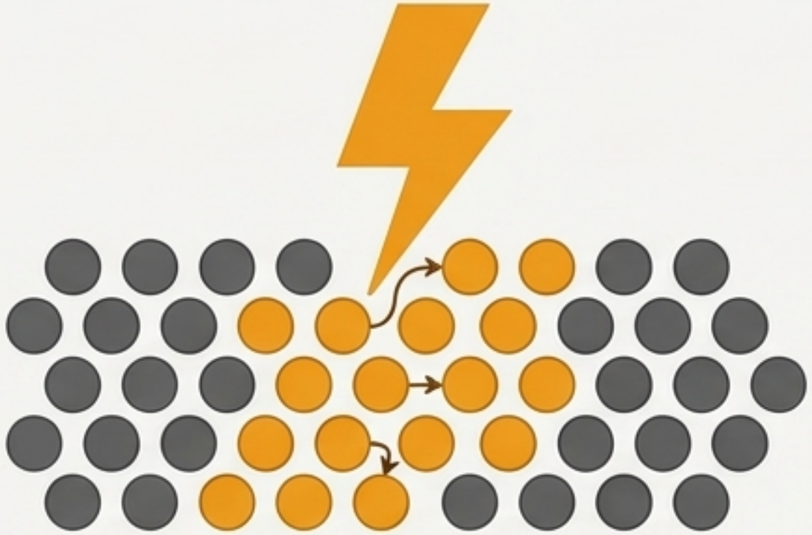
Deduction: A single node successfully gained disproportionate influence over the outcome. While it didn't flip the verdict here, in a closer election, such an attack would be decisive.

System Limitation: The smart contract fails to enforce a 'one vote per authorized identity per round' constraint. It lacks protection against replay or duplication attacks within a single consensus session.

```
...
Node 50/50 vote#1: 1 -> block 1319 (Gas: 95066)

⚖️ ── CONSENSUS ──
🗳️ Votes for 0: 12
🗳️ Votes for 1: 39
🏆 Final verdict: CLASSE 1
⚪ Duration: 1.19s | Votes counted (including Sybil): 51
👁️ Votes detail (first 50): [0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0,1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 0, 1, 1, 1, 1, 0]
```

Attack 5: Breaking the Bagging Assumption with Correlated Failure



Attack Mechanism:

A common distortion is introduced into the input data for many nodes (`--mask-top-k 5`). This forces many trees to make the same mistake, violating the core principle of bagging, which relies on *uncorrelated* errors to reduce variance.

Baseline Margin:

Margin = 28

Correlated Failure Margin:

Margin = 6

Experiment: Correlated Input Error

Command: `... --force-class 1 --correlated-failure --mask-top-k 5`

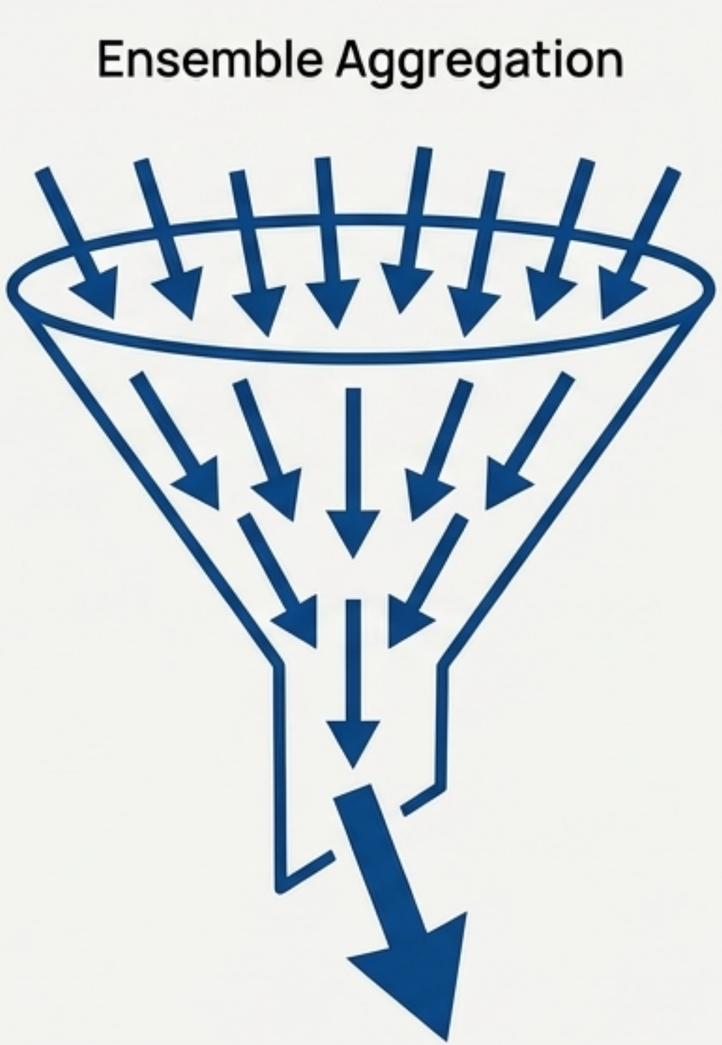
- **Votes (0):** 22
- **Votes (1):** 28
- **Verdict: CLASSE 1** (Correct) ✅
- **Margin:** 6 (vs. 28 in Baseline)

Deduction: The safety margin collapses from 28 to just 6. The system becomes extremely fragile and vulnerable to even a small number of malicious nodes flipping their votes.

System Limitation: The ensemble's robustness is predicated on error diversity. Systematic flaws in data preprocessing or widespread feature corruption can lead to a catastrophic, monolithic failure of the entire ensemble.

```
--- CONSENSUS ---
Votes for 0: 22
Votes for 1: 28
Final verdict: CLASSE 1
Duration: 1.13s | Votes counted (including Sybil): 50
Votes detail (first 30): [0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0]
```

Attack 6: The Danger of Systematic Bias Amplification



Attack Mechanism:

A small, systematic probability bias is added to every node's prediction before the final vote is cast (`--bias-shift 0.4`). This simulates a subtle, underlying flaw in the model or data that affects all participants equally.

Experiment: Systematic Probability Shift

Command: `... --force-class 1 --bias-shift 0.4 --bias-threshold 0.6`

- **Votes (0):** 11
- **Votes (1):** 39
- **Verdict:** CLASSE 1 (Correct) ✅
- **Margin:** 28

Deduction: While the verdict did not flip in this specific case, the mechanism demonstrates a critical vulnerability. An ensemble will not reduce bias; it may even amplify it. A systematic flaw affecting all nodes is not averaged out.

System Limitation: Bagging is effective at reducing variance, not bias. If the base learners are collectively biased, the ensemble will inherit and reflect that bias. The system has no mechanism to detect or correct for it.

```
⚖️ --- CONSENSUS ---
🗳️ Votes for 0: 11
🗳️ Votes for 1: 39
🏆 Final verdict: CLASSE 1
```

Findings & The Path to a Hardened System

Our experiments reveal a clear picture of the system's capabilities and its fundamental limitations.

Strengths

- **Transparency & Auditability:** The blockchain provides an immutable, public ledger of every vote, enabling full decision traceability.
- **Resilience to Minority Attacks:** The ensemble approach effectively tolerates a significant minority of faulty or malicious nodes.
- **Conceptual Simplicity:** The system is straightforward to implement, combining two mature technologies.

Weaknesses

- **Vulnerability to Majority Corruption:** The system is inherently insecure if more than 50% of voting power becomes malicious.
- **Naive Protocol Security:** The current smart contract is vulnerable to protocol-level attacks like Sybil and Replay.
- **Sensitivity to Correlated Failures:** Robustness depends on the assumption of independent node errors, which can be violated.

Proposed Improvements

- **Weighted & Staked Voting:** Tie voting power to a node's reputation or a financial stake (e.g., Proof-of-Stake) to disincentivize malicious behavior.
- **Identity Management & Nonces:** Implement an on-chain registry of authorized nodes (e.g., Decentralized Identifiers) and use nonces to prevent vote replays.
- **Fault Detection & Diversity:** Introduce mechanisms to detect correlated data inputs and consider using a more diverse set of models, not just trees from the same forest.