# Model Update Detection using History of Gradients

* Defense mechanism designed for federated learning systems to detect and differentiate between malicious clients (those intentionally sending harmful updates) and unreliable clients (benign clients with low-quality data).
* clients.py: Behaviour of standard client. Each client trains the global model on its local data and generates gradient updates to be sent back to the server.
* clients\_attackers.py: Malicious clients. These intentionally manipulate their gradient updates to degrade the performance of the global model. The implemented attack types include:
  + Sign-flipping attacks: Inverting the sign of gradients to mislead model training.
  + Additive-noise attacks: Adding noise to gradients to introduce errors.
  + Label-flipping attacks: Training on data with incorrect labels to corrupt the model.
  + Multi-label-flipping attacks: Flipping multiple labels to target specific classes.
* server.py: It incorporates the MUD-HoG algorithm to analyse the history of gradients from each client, identifying and differentiating between malicious and unreliable clients.
  + Detect malicious clients and exclude or mitigate their impact.
  + Identify unreliable clients and adjust their contributions accordingly.
  + Distinguish between targeted and untargeted attacks among malicious clients.

### **Key Stages of MUD-HoG**

MUD-HoG identifies malicious and unreliable clients in a series of steps:

#### **Stage 1: Collect Gradients**

* For each client:
  + **sHoG (short-term)**: Recent updates normalized by their L2 norm.
  + **lHoG (long-term)**: Cumulative gradient history.

#### **Stage 2: Detect Attacks**

##### **(a) Flip-Sign Attack Detection**

* **Goal**: Detect clients reversing gradient directions to mislead the global model.
* **Method**:
  + Compute the **cosine similarity** of each client’s normalized sHoG with the median gradient of all clients.
  + If the similarity is negative (angle > 90°), the client is flagged as a **flip-sign attacker**.

##### **(b) Untargeted Attack Detection**

* **Goal**: Detect clients adding noise or random gradients.
* **Method**:
  1. Exclude flip-sign attackers.
  2. Use **DBSCAN clustering** on the raw sHoG gradients:
     + Largest cluster represents normal clients.
     + Outliers are flagged as **untargeted attackers**.

##### **(c) Targeted Attack Detection**

* **Goal**: Identify clients attempting to manipulate the model subtly towards specific outcomes.
* **Method**:
  1. Exclude flip-sign and untargeted attackers.
  2. Use **KMeans clustering** (K=2) on the lHoG gradients.
     + Minority cluster is flagged as **targeted attackers**.

##### **(d) Unreliable Clients Detection**

* **Goal**: Detect clients with noisy or inconsistent updates.
* **Method**:
  + After excluding malicious clients:
    1. Compute the **cosine similarity** of each sHoG to the median gradient.
    2. Clients with small cosine similarities (large angles) are flagged as **unreliable**.