Algorithm 1 GuardedLearn (GL)

Require: Data set St, threshold λ, data partitions M, N, and learning rate r
Ensure: Benign cluster

```
Divide the data set M vertically into n partitions.
Distribute partition M_j to participant u_j.
Initialize an empty set G.
Set θ_{t−1} as the global model parameters after the previous training round t − 1.
for each neuron i in the final layer |C| do
    for each partition j in St do
        Update parameters θ_{t,j} after training.
        Compute the parameter difference θ_{Δ,j} = θ_{t,j} − θ_{t−1}.
        Extract parameters θ_{iΔ,j} connected to neuron i in the final layer.
        Convert θ_{iΔ,j} to a numpy array X(i)_j.
        Transform X(i)_j to X'(i)_j = M_j X(i)_j N.
        Add partition M_j to G.
    end for
    Construct matrix Y'(i) = [X'(i)_1, X'(i)_2, ..., X'(i)_n].
    Perform Singular Value Decomposition (SVD) on Y'(i) to obtain U'_i, Σ'_i, and V'^T_i.
    Calculate U_i = G^T U'_i.
    Compute bY_i = U_i Σ'_i.
    Apply Algorithm 2 to bY_i to classify benign weights.
end for
Pass benign updates bY_i to Algorithm 3 (RFA) for aggregation.
```

Algorithm 2 Clustering using DBSCAN

Require: Data matrix M, threshold ϵ, minimum points $MinPts$
Ensure: Cluster labels

Initialize an empty list $clusters$
Initialize an empty set $visited$
for each data point p in M **do**
  if p is visited **then**
    Continue to the next data point
  end if
  Mark p as visited
  $NeighborPts \leftarrow$ regionQuery(p, ϵ)
  if size($NeighborPts$) $<$ $MinPts$ **then**
    Mark p as noise
  else
    Create a new cluster C and add p to C
    ExpandCluster(M, p, $NeighborPts$, C, ϵ, $MinPts$, $visited$)
    Add C to $clusters$
  end if
end for
return Cluster labels

Algorithm 3 Robust Federated Aggregation (RFA) [1]

Require: Initial parameter vector $w^{(0)}$, total communication rounds T, clients per round m, local update iterations τ, step size γ, convergence threshold ε

for $t = 0, 1, \ldots, T-1$ **do**
  Randomly select m clients from the cluster of benign clients
  for each selected client i in parallel **do**
    Set initial local parameter vector $w^{(t)}i, 0 = w^{(t)}$
    for $k = 0, \ldots, \tau - 1$ **do**
      Sample data batch $z^{(t)}i, k \sim D_i$
      Update local parameter: $w^{(t)}i, k+1 = w^{(t)}i, k - \gamma \nabla f(w^{(t)}i, k; z^{(t)}i, k)$
    end for
    Set global parameter: $w^{(t+1)}i = w^{(t)}i, \tau$
  end for
  Perform federated aggregation: $w^{(t+1)} = GM\left(w^{(t+1)}i\right) i \in St, (\alpha_i)_{i \in St}, \varepsilon$ (Refer Algo. 2)
end for
return Updated global parameter vector $w^{(T)}$

Bibliography

[1] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.