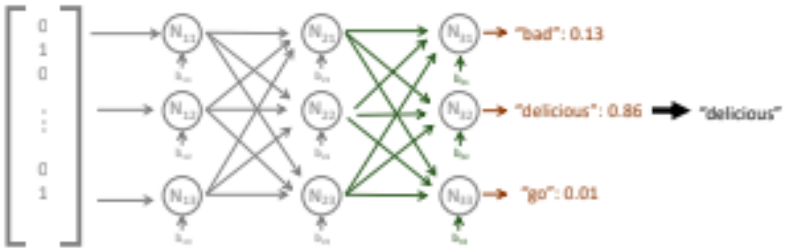
DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection



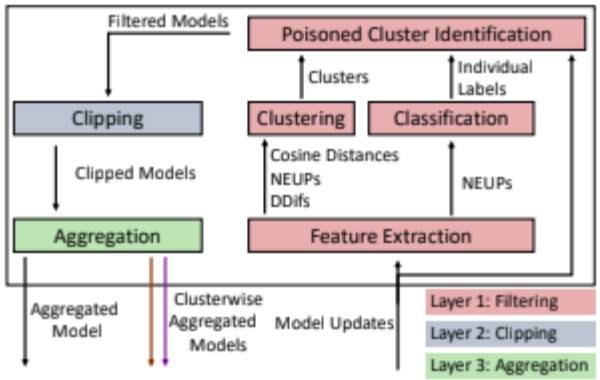


Fig. 2: Structure of DeepSight

Algorithm 1 Filtering Layer

Input: N, ▷ number of models
W, ▷ list of N received local models
G_t ▷ global model
Parameters: τ, ▷ Threshold of suspicious models for excluding cluster
seeds, ▷ 3 seeds for generating random data for ddfis
input_dim ▷ dimension of a single input
Output: accepted_models
▷ Feature Extraction
cosine_distances $\leftarrow 0^{N \times N}$
global_bias $\leftarrow$ output_layer_bias(G_t)
for each clients i, j in $[1, N]$ **do**
    update$_i$ $\leftarrow$ output_layer_bias(W_i) - global_bias
    update$_j$ $\leftarrow$ output_layer_bias(W_j) - global_bias
    cosine_distances$_{i,j}$ $\leftarrow$ 1 - COSINE(update$_i$, update$_j$)
end for
$\forall i \in \{1, \dots N\}$: neups$_i$ $\leftarrow$ NEUPs(G_t, W_i)
$\forall i \in \{1, \dots N\}$: thresh_exds$_i$ $\leftarrow$ THRESHOLD_EXCEEDING(neups)
$\forall i \in \{1, 2, 3\}$: rand_input_data$_i$ $\leftarrow$ random_matrix(seeds$_i$, 20 000, input_dim)
$\forall i \in \{1, 2, 3\}$: ddifs$_i$ $\leftarrow$ DDIFs(rand_input_data$_i$, G_t, $W_1 \dots W_n$)
▷ Classification
classificat_boundary $\leftarrow$ MEDIAN(thresh_exds) / 2
$\forall i \in \{1, \dots N\}$: labels$_i$ $\leftarrow$ (thresh_exds[i] $\leq$ classificat_boundary)? 1:0
▷ Clustering
clusters $\leftarrow$ CLUSTER(N, neups, ddifs, cosine_distances)
▷ PCI
accepted_models $\leftarrow \{\}$
for cluster in clusters **do**
    amount_of_positives $\leftarrow$ SUM(labels[cluster]) / |cluster|
    if amount_of_positives $< \tau$ **then**
        accepted_models $\leftarrow$ accepted_models $\cup$ models[cluster]
    end if
end for

Algorithm 2 Clustering

procedure DISTSFROMCLUST(clusters, N)
    $\forall i, j \in \{1, \dots N\}$: pairwise_dists$_{i,j}$ ← cluster_of_model(i, clusters) == cluster_of_model(j, clusters)? 0:1 ▷ cluster_of_model(x, clusters) returns the cluster that contains the model with index x
    return pairwise_dists
end procedure

Input:
N, ▷ N is the number of models
neups, ▷ NEUPs as list of N vectors with dimension P
ddifs ▷ DDifs as list of 3 lists of vectors with dimension P
cosine_distances ▷ cos_distances a matrix $\in \mathbb{R}^{N \times N}$
Output: clusters ▷ clusters as set of sets of indices

cosine_clusters ← HDBSCAN(distances = cosine_distances)
cosine_cluster_dists ← DistsFromClust(cosine_clusters, N)
neup_clusters ← HDBSCAN(values = neups)
neup_cluster_dists ← DistsFromClust(NEUP_clusters, N)
$\forall i \in \{1, 2, 3\}$: ddif_clusters$_i$ ← HDBSCAN(values = ddifs$_i$)
$\forall i \in \{1, 2, 3\}$: ddif_clust_dists$_i$ ← DistsFromClust(ddif_clusters$_i$,N)
merged_ddif_clust_dists ← AVG(ddif_clust_dists$_1$, ddif_clust_dists$_2$, ddif_clust_dists$_3$)
 ▷ Combine clusterings
merged_distances ← AVG(merged_ddif_clust_dists, neup_clust_dists, cosine_clust_dists)
clusters ← HDBSCAN(distances = merged_distances)