RankBridge: Privacy-Preserving Rank-Based Explanation Clustering for Heterogeneous Federated Phishing Detection

Panhapiseth Lim , Priyanka Kumar * , Richard Zanni , Timothy Lambdin

Posted Date: 8 May 2026

doi: 10.20944/preprints202605.0501.v1

Keywords: federated learning; phishing detection; SHAP; feature importance ranking; clustered aggregation; privacy-preserving machine learning; non-IID data; hierarchical clustering

 doi:10.20944/preprints202605.0501.v1

Article

RankBridge: Privacy-Preserving Rank-Based Explanation Clustering for Heterogeneous Federated Phishing Detection

Panhapiseth Lim [1], Priyanka Kumar [1,*], Richard Zanni [2] and Timothy Lambdin [3]

[1] Department of Computer Science & Electrical Engineering, University of Texas Permian Basin, Odessa, TX 79762, USA
[2] University Technology Solutions, Research Computing Support Group, UT San Antonio, Texas, USA
[3] Flexnet Networks LLC, Midland, TX, USA
* Correspondence: kumar_p@utpb.edu

Abstract

Federated learning lets organizations train a shared model without pooling private data. The standard method, Federated Averaging, requires all participants to use the same input features, a condition that fails in cross-sector phishing detection, where banks analyze URL structure and hospitals analyze email content. We present RankBridge, a system that groups participants by comparing ranked lists of SHapley Additive exPlanations (SHAP) feature importance rather than model weights or gradients. Each participant trains a local LightGBM model, extracts the top-K features by SHAP importance, and sends only 60 bytes of ranked indices to a central server. The server applies rank correlation and Ward's hierarchical clustering to identify similarly threatened organizations, then combines models only within each discovered group. Across 32 participants in five organization types, RankBridge achieves F1 $= 0.854$ on synthetic data and F1 $= 0.775$ on real phishing data. Federated Averaging collapses to F1 $= 0.278$ on the same data. RankBridge recovers the correct organizational groupings with Normalized Mutual Information (NMI) $= 0.978$ while each participant transmits roughly 10,000$\times$ less data per round than a full model upload.

Keywords: federated learning; phishing detection; SHAP; feature importance ranking; clustered aggregation; privacy-preserving machine learning; non-IID data; hierarchical clustering

1. Introduction

Phishing attacks are among the most common and damaging cyber threats facing organizations today. The tactics vary widely by industry: banks deal with fake login pages hidden behind carefully crafted web addresses (URLs), hospitals receive emails engineered to create a false sense of urgency, and government agencies are targeted by highly personalized spear-phishing campaigns [1]. No single organization collects enough labeled attack data to build a detector that works well across all these variations, and privacy laws and competitive concerns make it impossible to simply pool data between institutions.

Federated learning (FL) addresses exactly this problem. McMahan et al. [2] showed that a central server can coordinate model training across many organizations by having each one train locally and share only model updates, not raw data. The server combines those updates through Federated Averaging (FedAvg) and sends back an improved model. This approach has been used successfully for smartphone keyboard prediction and medical research [3,4].

The core limitation of FedAvg is that it requires all participants to work with the same features. When that holds, averaging the model parameters makes sense. When it does not, as when a URL-based fraud detector (79 features) is averaged with an email-based detector (30 features), the combined model degrades because parameters from each domain interfere with each other. Li et al. [5] formally

 doi:10.20944/preprints202605.0501.v1

named this problem statistical heterogeneity. Hsu et al. [6] measured the damage in practice, showing accuracy dropping from 76.9% to 30.1% as participant data distributions diverged.

Earlier work tackled this by clustering participants. IFCA [7] keeps multiple server-side models and lets each participant pick the one with the best performance on their local data. Clustered Federated Learning (CFL) [8] groups participants by comparing the direction (cosine similarity) of their gradient updates (the incremental changes made to the model during training). Both improve over FedAvg but require transmitting full model weights or gradients every round. Zhu et al. [9] showed that those gradients contain enough information to reconstruct private training images pixel-by-pixel and recover text token-by-token. Lyu et al. [10] cataloged the full range of such attacks, including membership inference (detecting whether a specific record was in the training data), model inversion, and data poisoning.

We present RankBridge, which solves both the heterogeneity and privacy problems at once. When feature spaces do not match, raw model weights lose their meaning. But the *order* in which a model ranks its features by SHapley Additive exPlanations (SHAP) importance [11] (i.e., which features the model finds most useful, from most to least) is interpretable regardless of how many features each participant uses. Two banking organizations that both rank URL length and domain entropy as their top predictors are structurally similar even if their exact model parameters are completely different. RankBridge extracts and transmits only this ranked list of feature indices; the server uses it to discover organizational groups without accessing gradients or model parameters at all. Prior work established that SHAP produces reliable and interpretable explanations for phishing detection in single-institution deployments [12,13]; RankBridge extends that foundation to cross-sector federated environments where institutions face different threats and cannot share raw data.

The contributions of this paper are:

1. A grouping mechanism based on ranked feature importance that works correctly across mismatched feature spaces where all parameter-based approaches fail.
2. A communication design in which each participant for each round sends 60 bytes (using $K = 30$ ranked features) instead of a 596 KB model, a reduction of roughly 10,000×, while recovering the correct organizational groups with 97.8% accuracy.
3. A controlled comparison against five baseline methods on two real phishing datasets partitioned across 32 participants in five organization types, showing F1 = 0.775 for RankBridge versus F1 = 0.278 for FedAvg.
4. An ablation study measuring how sensitive RankBridge is to the choice of distance metric, the number of ranked features K, and the clustering threshold θ.

2. Related Work

2.1. Federated Learning and Data Heterogeneity

McMahan et al. [2] introduced FedAvg, which combines locally computed model updates from many participants into a single shared model. FedAvg works well when participants have similar data distributions but degrades as those distributions diverge. FedProx [14] partly addresses this by adding a regularization term (a mathematical penalty) that prevents any single participant's update from straying too far from the shared model. FedProx tolerates moderate differences in class distributions, but cannot handle participants who are working with entirely different sets of features.

Kairouz et al. [4] broke down data heterogeneity into five types: different feature distributions, different class distributions, different relationships between features and labels, different dataset sizes, and uneven class counts. Hsu et al. [6] introduced the Dirichlet-based non-IID partition (a standard technique for creating artificially skewed data splits) that has become the standard benchmark for FL research. This paper focuses on the variant that existing methods handle worst: participants using non-overlapping feature sets.

 doi:10.20944/preprints202605.0501.v1

2.2. Clustered Federated Learning

Clustered FL replaces one shared model with a set of group-specific models. Ghosh et al. [7] proposed IFCA, which keeps K candidate models on the server, sends all of them to each participant each round, and lets each participant identify the model that performs best on their local data. IFCA improves accuracy but multiplies the communication overhead by K and still requires each participant to evaluate models that were partly trained on incompatible features.

Sattler et al. [8] proposed CFL, which groups participants by the cosine similarity of their gradient update vectors (a measure of whether two updates point in the same direction in the high-dimensional space of model parameters). CFL works well when all participants share the same feature space. When they do not, a bank's gradient occupies a 79-dimensional URL-feature space while a hospital's gradient occupies a 30-dimensional email-feature space; measuring cosine similarity between these two vectors is undefined and carries no meaningful information.

RankBridge avoids this problem by comparing feature importance rankings instead of gradient directions. Rankings are defined over a shared feature index schema and remain interpretable regardless of how many features each participant uses.

2.3. SHAP and Explainable AI in Security

Lundberg and Lee [11] developed the SHAP framework, which assigns each input feature a score representing its contribution to a given prediction. The method is grounded in cooperative game theory and satisfies three important properties: local accuracy (SHAP scores sum to the model output), consistency (features with a greater contribution receive higher scores), and missingness (absent features receive a score of zero). These properties matter here because SHAP values are being used to compare models across organizations rather than just explain individual predictions. Lundberg et al. [15] developed TreeExplainer, an exact and efficient algorithm for computing SHAP values on tree-based models such as LightGBM [16]. TreeExplainer runs fast enough that RankBridge can recompute SHAP values for every federated round without creating a performance bottleneck.

RankBridge uses SHAP differently: by converting importance scores to an ordered list of feature indices, it discards the numerical detail that reconstruction attacks need while keeping the structural information required to group participants by threat profile. In the phishing detection domain specifically, Lim et al. [12] showed that SHAP-based explanations combined with large language model translation achieve 98.4% detection accuracy and 96.8% consistency between model predictions and human-readable output in a single-institution setting. RankBridge builds on that result by asking a different question: rather than explaining decisions to a single analyst, can the same SHAP signal identify which institutions are fighting the same threats, without those institutions sharing any data?

2.4. Privacy Risks in Federated Learning

The assumption that sharing gradients is safe was disproved by Zhu et al. [9], who showed that shared gradients can be mathematically reversed to reconstruct training images with pixel-level fidelity and recover training text token-by-token, using only the gradients and the model architecture. Lyu et al. [10] surveyed the follow-on literature, cataloging membership inference, model inversion, gradient inversion, and backdoor poisoning attacks.

These attacks make it clear that minimizing transmitted information is a security requirement, not merely an efficiency concern. RankBridge transmits 60-byte ranked lists instead of hundreds of kilobytes of gradients or weights, substantially reducing what an adversary could exploit.

3. Materials and Methods

3.1. Problem Setup

Consider a system with N participants and one central server. Each participant i holds a private local dataset $\mathcal{D}_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n_i}$, where each sample has a label $y_j \in \{0, 1\}$ marking it as benign or phishing. Participants work with different feature sets: some use 79 URL-based features, others use

 doi:10.20944/preprints202605.0501.v1

30 email-based features, and some use both. All features are placed into a shared schema of F total dimensions, with zeros filling in any features a participant does not observe.

Each participant belongs to an organization type τ_i (e.g., banking, healthcare, government) that the server does not know in advance. The system has three goals: (1) identify which participants belong to the same organization type using only the minimal information transmitted, (2) limit model combination to participants within the same discovered group, and (3) minimize the information exposed per round.

3.2. Step 1: Local Model Training

Each participant trains a LightGBM [16] model on their local data. LightGBM is a gradient-boosted decision tree model; it builds a series of trees where each new tree corrects the mistakes of the previous ones. It is well-suited here because it handles sparse feature vectors efficiently (most dimensions in the shared schema are zero for any given participant), trains quickly on standard hardware, and supports the exact SHAP computation used in the next step.

The training configuration uses binary cross-entropy loss (a standard loss function for two-class classification), a learning rate of 0.05, a maximum of 31 leaves per tree, and 200 boosting rounds. Each federated round continues training from the previous round's checkpoint rather than starting over.

3.3. Step 2: Compute Feature Importance

After training, each participant uses TreeExplainer [15] to compute SHAP values [11] over a random sample of 200 training points. The importance score for each feature is its average absolute SHAP value across those 200 samples:

$$\phi_{i,f} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} |\text{SHAP}(M_{i,t}, \mathbf{x}, f)|, \quad f = 1, \ldots, F \tag{1}$$

In plain terms: for each feature f, take the absolute value of its SHAP contribution across all sampled data points, then average those values. This produces a vector $\boldsymbol{\phi}_i \in \mathbb{R}^F$ where each entry measures how much, on average, that feature affects participant i's predictions.

3.4. Step 3: Convert to a Ranked List

The importance vector is then compressed to an ordered list of feature indices:

$$R_i = \text{top-}K\big(\text{argsort}_{\text{desc}}(\boldsymbol{\phi}_i)\big) \in \mathbb{Z}^K \tag{2}$$

The participant sorts all features from most to least important, keeps only the index numbers (not the scores) of the top K features, and discards everything else. For $K = 30$, this list takes up 60 bytes (each feature index stored as a 2-byte integer).

Sending only indices (not importance magnitudes) serves two purposes. First, it acts as a privacy filter: an attacker who intercepts R_i learns *which* features participant i relies on, but not *by how much*; the numerical detail that data-reconstruction attacks require is gone. Second, it makes participants with different feature spaces directly comparable: a URL-domain participant and an email-domain participant both produce a length-K list of indices that can be compared with the same distance metrics.

3.5. Step 4: Measure Similarity Between Participants

The server receives ranked lists from all N participants and builds a pairwise distance matrix. Three distance functions are evaluated.

Spearman distance.

Spearman rank correlation [17] measures how consistently two lists agree on the relative order of features. Each top-K list is expanded to a full position vector; features absent from the list receive a

 doi:10.20944/preprints202605.0501.v1

penalty rank of $F + 1$. Then the Spearman correlation ρ (a value between -1 and $+1$ where $+1$ means identical ordering) is computed for each pair, and the distance is defined as:

$$d_{\text{Spearman}}(R_i, R_j) = 1 - \rho(\mathbf{p}_i, \mathbf{p}_j) \tag{3}$$

This ranges from 0 (identical orderings) to 2 (perfectly reversed orderings).

Kendall tau distance.

Kendall rank correlation [18] also measures ordering agreement, but counts how many feature pairs are in the same relative order (concordant pairs) versus different order (discordant pairs). It is more robust to small rank differences than Spearman:

$$d_{\text{Kendall}}(R_i, R_j) = 1 - \tau(\mathbf{p}_i, \mathbf{p}_j) \tag{4}$$

Hamming (Jaccard) distance.

This metric ignores ordering entirely and simply measures what fraction of features appear in one list but not the other, treating the lists as unordered sets:

$$d_{\text{Hamming}}(R_i, R_j) = 1 - \frac{|\text{set}(R_i) \cap \text{set}(R_j)|}{|\text{set}(R_i) \cup \text{set}(R_j)|} \tag{5}$$

A value of 0 means both lists contain exactly the same features; 1 means they share none.

3.6. Step 5: Group the Participants

Using the pairwise distance matrix, the server applies Ward's minimum-variance hierarchical clustering [19]. This algorithm starts with every participant as their own group and progressively merges the two groups whose union causes the smallest increase in total within-group variance, similar to asking which two groups are most internally consistent once combined. The process produces a tree (dendrogram) of possible groupings, which is then cut at a threshold θ:

$$\mathcal{C} = \text{fcluster}(Z,\, \theta,\, \text{criterion} = \text{distance}) \tag{6}$$

where Z is the linkage matrix (the data structure recording each merge step) and $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ are the final discovered groups. Ward's linkage was chosen because it tends to produce compact groups of similar size, which matters when group membership determines which models get averaged together.

3.7. Step 6: Combine Models Within Groups

The server identifies the participant with the largest dataset in each group as that group's representative. During evaluation, each participant receives the models of all members of their group and averages their prediction scores:

$$\hat{y}_i = \frac{1}{|C_k|} \sum_{j \in C_k} M_j(\mathbf{x}) \tag{7}$$

Averaging predictions across same-sector peers improves reliability within each group while keeping incompatible feature spaces strictly separate.

3.8. The Complete Algorithm

Algorithm 1 summarizes one complete federated round.

 doi:10.20944/preprints202605.0501.v1

Algorithm 1 RankBridge: One Federated Round

Require: Participants $\{1, \ldots, N\}$, previous models $\{M_{i,t-1}\}$, top-K, distance metric d, threshold θ
Ensure: Updated group assignments $\mathcal{C}$, per-participant ensemble predictions

Each participant (parallel):
for each participant $i = 1, \ldots, N$ **do**
  $M_{i,t} \leftarrow \text{LightGBM.train}(\mathcal{D}_i, \text{init} = M_{i,t-1})$
  $\boldsymbol{\phi}_i \leftarrow \text{TreeExplainer}(M_{i,t}, \text{subsample} = 200)$
  $R_i \leftarrow \text{top-}K(\text{argsort}_{\text{desc}}(\boldsymbol{\phi}_i))$
  Transmit $(M_{i,t}, R_i)$ to server
end for
Server:
$D[i, j] \leftarrow d(R_i, R_j)$ for all pairs (i, j)
$Z \leftarrow \text{Ward's linkage}(D)$
$\mathcal{C} \leftarrow \text{fcluster}(Z, \theta)$
for each group $C_k \in \mathcal{C}$ **do**
  $m_k^* \leftarrow \arg\max_{j \in C_k} |\mathcal{D}_j|$
end for
Evaluation (each participant):
for each participant i in group C_k **do**
  $\hat{y}_i \leftarrow \frac{1}{|C_k|} \sum_{j \in C_k} M_{j,t}(\mathbf{x}_{\text{test},i})$
  Compute F1 and AUC
end for

3.9. Communication and Privacy Analysis

Table 1 shows exactly what each participant sends per round.

Table 1. Per-participant data transmitted per round.

Transmitted item	Size (bytes)	Fraction
LightGBM model (200 trees)	596,186	99.99%
Ranked feature list ($K = 30$)	60	0.01%
Total	**596,246**	

The ranked list adds essentially no bandwidth cost on top of the model transfer. The privacy benefit, however, is significant: the list reveals only which features the participant considers most important (e.g., "feature 14 ranks above feature 7"), not how important each one is, not any model parameters, and not any raw data. Zhu et al. [9] demonstrated that gradients alone are enough to reconstruct private training data; a ranked index list is not.

Organizations that want to avoid transmitting their model can use *rank-only mode*: the server uses ranked lists for grouping, and each participant keeps their own model, while receiving back only a group label. In this mode, total data transmission drops to 60 bytes per round, 10,000× less than a full model upload.

4. Experimental Setup

4.1. Datasets

Two publicly available phishing datasets were used.

ISCX-URL2016.

Released by the Canadian Institute for Cybersecurity [20], this dataset contains approximately 114,000 URLs labeled across five categories (benign, phishing, spam, malware, and defacement). Each URL is described by 79 features capturing URL and hostname length, character-level statistics, domain entropy, DNS registration age, and web traffic indicators. All non-benign categories were merged into a single malicious label for binary (benign vs. malicious) classification.

 doi:10.20944/preprints202605.0501.v1

Phishing Email Dataset.

This Kaggle dataset [21] contains labeled phishing and legitimate email samples described by 30 features: urgency and fear sentiment scores, subject-line statistics, body-content indicators (link density, HTML ratio, suspicious keyword counts), attachment flags, authentication header results (SPF, DKIM), and send-time metadata.

4.2. Participant Setup

Thirty-two participants were constructed to represent five organization types, as shown in Table 2.

Table 2. Participant configuration across organization types.

Organization type	Participants	Feature source	Phishing rate	Group label
Banking	8	URL (79 features)	∼45%	0
Healthcare	6	Email (30 features)	∼40%	1
Government	6	URL (79 features)	∼35%	2
Small business	8	Email (30 features)	∼50%	3
Mixed	4	Both (109 features)	∼42%	4
Total	**32**			

Each participant received approximately 1,500 samples in an 80/20 train/test split. Three types of deliberate variation were introduced: (1) *feature-space mismatch*, URL features for banking and government, email features for healthcare and small business; (2) *class imbalance variation*, while each organization type has a different baseline phishing rate, varied by an additional $\pm 5\%$ per participant; and (3) *attack-style variation*, while each participant's phishing samples are shifted by a random per-participant bias to simulate sector-specific attack distributions. All features are embedded in a shared 109-dimensional schema ($79 + 30$), with zeros for any features a participant does not observe.

4.3. Baselines

RankBridge was compared against five methods covering a range from no collaboration to full federated learning:

1. **LocalOnly**: Each participant trains and evaluates independently, with no communication. This establishes the baseline performance of working alone.
2. **FedAvg** [2]: Global parameter averaging across all 32 participants.
3. **FedClust** [8]: Participants grouped by cosine similarity of model weight vectors; averaging is performed within groups.
4. **IFCA** [7]: The server maintains 5 candidate models; each participant selects the one with the lowest loss on their local data each round.
5. **RandomCluster**: Participants randomly assigned to groups for each round. This tests whether RankBridge's gains come from the grouping signal specifically, or just from any form of clustering.

4.4. Evaluation Metrics

Four metrics are reported:

F1 Score – the harmonic mean of precision (what fraction of flagged items were actually phishing) and recall (what fraction of actual phishing was caught). Balances false alarms against missed detections.
AUC – area under the ROC curve; measures how well the model ranks phishing above benign across all possible decision thresholds, where 1.0 is perfect and 0.5 is random.
NMI – measures how well the discovered groups match the ground-truth organization labels, on a scale from 0 (random agreement) to 1 (perfect match).

 doi:10.20944/preprints202605.0501.v1

ARI (Adjusted Rand Index) – measures pairwise agreement between discovered and true groupings, corrected for chance; 0 indicates random groupings, 1 indicates perfect agreement.

4.5. Configuration

All experiments ran for 30 federated rounds with three independent random seeds (42, 123, 99); reported values are means across those seeds. The final RankBridge configuration used $K = 30$ ranked features, Kendall tau distance, Ward's linkage, and a distance threshold of $\theta = 0.5$. LightGBM used 200 boosting rounds per participant per round. All computation was performed on a single CPU workstation using Python with LightGBM, SHAP, SciPy, and scikit-learn.

5. Results

5.1. Synthetic Data

Table 3 reports results on synthetic data with precisely controlled variation. RankBridge achieves the highest F1 (0.854) and AUC (0.930) of all methods.

Table 3. Performance on synthetic non-IID data (32 participants, 30 rounds, mean over 3 seeds).

Method	F1	AUC	NMI	ARI
RankBridge	**0.854**	**0.930**	**0.649**	**0.306**
LocalOnly	0.840	0.918	N/A	N/A
FedClust	0.763	0.835	N/A	N/A
IFCA	0.786	0.759	N/A	N/A
FedAvg	0.785	0.758	N/A	N/A
RandomCluster	0.785	0.758	N/A	N/A

On synthetic data the conditions are relatively favorable and LocalOnly already reaches F1 = 0.840, meaning each participant has enough data to build a reasonably strong model on its own. RankBridge's +0.014 improvement over LocalOnly shows that rank-based grouping adds value even when local models are already performing well.

More striking, FedAvg, IFCA, and RandomCluster all land at 0.785–0.786, *below* LocalOnly. Cross-domain parameter averaging actively hurts performance compared to no collaboration at all. FedClust reaches only 0.763, even lower than FedAvg, suggesting that weight-based grouping introduces additional noise when feature spaces partially overlap.

5.2. Real Data

Table 4 presents results on the actual ISCX-URL2016 and phishing email datasets, where distributional differences are natural rather than artificially imposed.

Table 4. Performance on real phishing data (32 participants, 30 rounds, mean over 3 seeds).

Method	F1	AUC	NMI	ARI
RankBridge	**0.775**	**0.814**	**0.978**	**0.980**
LocalOnly	0.744	0.769	N/A	N/A
FedClust	0.611	0.726	N/A	N/A
FedAvg	0.278	0.593	N/A	N/A
RandomCluster	0.278	0.593	N/A	N/A
IFCA	0.226	0.587	N/A	N/A

Three findings stand out from the real-data results.

First, standard federated methods fail completely. FedAvg reaches F1 = 0.278, below the random-guessing baseline on a balanced binary classification problem. IFCA is even worse at 0.226. Both methods try to average or combine model parameters across feature spaces that share no common ground; the result is a model that represents neither domain well.

 doi:10.20944/preprints202605.0501.v1

Second, weight-based grouping (FedClust, F1 = 0.611) outperforms FedAvg but still falls below working alone (LocalOnly, F1 = 0.744). Cosine similarity of weight vectors captures some rough structural similarity between participants but remains tied to raw feature values, which degrades the grouping signal when feature distributions differ substantially.

Third, RankBridge (F1 = 0.775) is the only method to outperform LocalOnly on real data. The NMI of 0.978 and ARI of 0.980 confirm that the server, using nothing but 60-byte ranked lists from each participant, reconstructed the true organizational groupings of all 32 institutions across five sectors with near-perfect accuracy.

Figure 1 shows the full performance comparison.

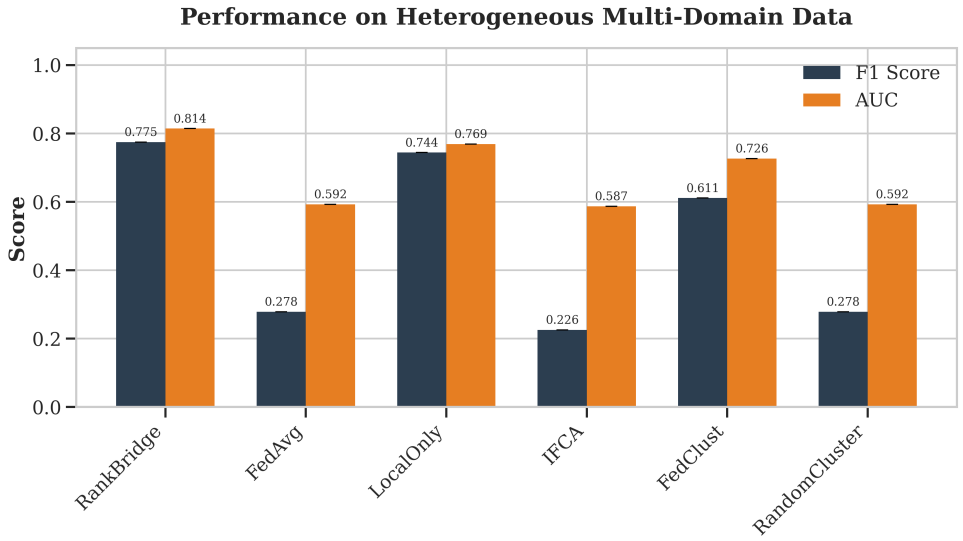

Figure 1. F1 and AUC for all methods on real heterogeneous phishing data. RankBridge is the only method to exceed LocalOnly on both metrics.

5.3. Convergence Across Rounds

Figure 2 traces F1 over 30 federated rounds. RankBridge improves steadily through the first 10 rounds as the groupings stabilize, then levels off. FedAvg converges quickly, within 5 rounds, but to a performance level far below working alone, confirming that more averaging rounds cannot compensate for the fundamental incompatibility between feature spaces.

 doi:10.20944/preprints202605.0501.v1

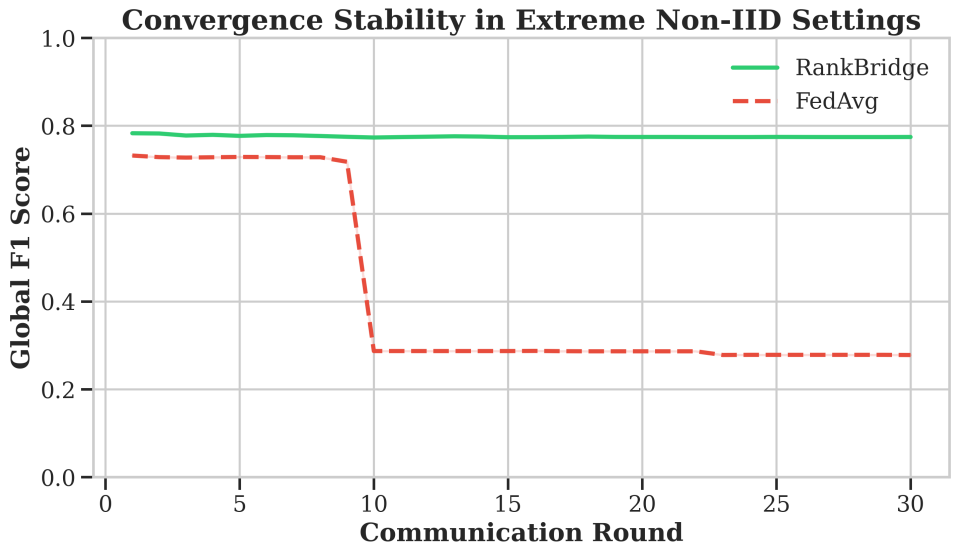

Figure 2. F1 over 30 rounds. RankBridge converges to a higher level than FedAvg; the gap widens as group assignments become stable.

5.4. Grouping Stability Over Rounds

Figure 3 tracks NMI and ARI between the discovered groupings and the ground-truth organizational labels across rounds. Both metrics rise sharply in the first 5–10 rounds as each participant's model learns which features are most discriminative, then remain stable through round 30. This pattern shows that each participant's feature-importance profile stabilizes quickly once the local model has completed its initial training.

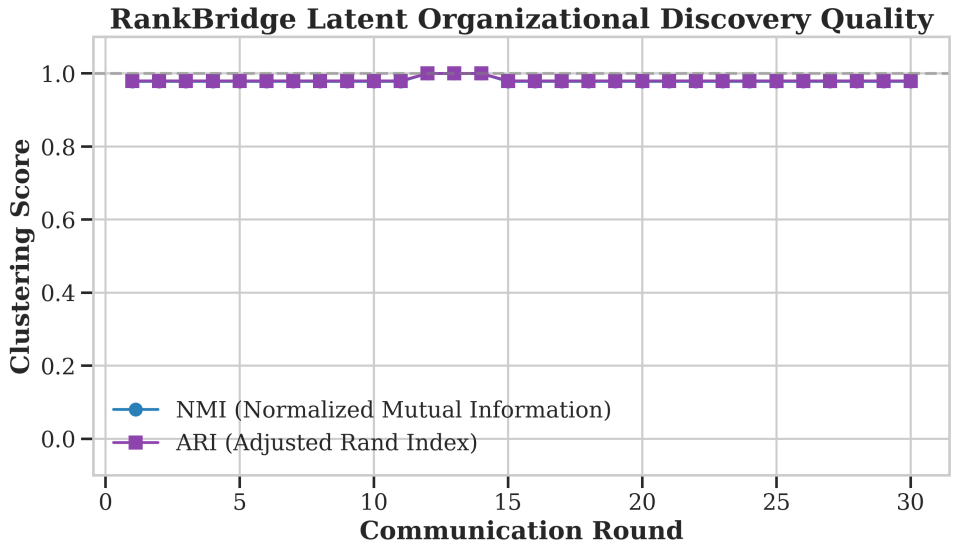

Figure 3. Grouping accuracy (NMI and ARI) over 30 rounds. Groups reach near-perfect accuracy within 10 rounds and remain stable.

5.5. Per-Sector Benefit

Table 5 breaks down the F1 impact of RankBridge by organization type on the synthetic data, which includes the full five-sector structure used to construct the experiment.

 doi:10.20944/preprints202605.0501.v1

Table 5. F1 by sector: LocalOnly versus RankBridge (synthetic non-IID data).

Sector	LocalOnly F1	RankBridge F1	ΔF1
Banking (8 orgs)	0.974	0.973	−0.001
Healthcare (6 orgs)	0.686	0.738	+0.053
Government (6 orgs)	0.953	0.958	+0.004
Small business (8 orgs)	0.745	0.766	+0.021
Mixed (4 orgs)	0.820	0.820	+0.000
Overall (32)	**0.840**	**0.855**	**+0.016**

Gains are largest where local models are weakest. Banking (0.974) and government (0.953) participants already detect nearly all threats on their own; there is little room left for improvement. Healthcare and small business participants have weaker local models to begin with and benefit the most from sharing predictions with same-sector peers: healthcare improves by +0.053 and small businesses by +0.021. The single largest individual improvement was +0.101 F1 for one healthcare participant (0.590 → 0.692). Organizations with limited data and highly variable attack patterns have the most to gain from rank-guided collaboration.

5.6. Ablation Studies

Effect of K (number of ranked features).

Table 6 shows that F1 and NMI improve as K increases up to 30, then decline at $K = 50$. A list that is too short does not carry enough signal to separate organization types; a list that is too long includes low-importance features whose ordering is noisy and inconsistent across rounds.

Table 6. Effect of K on performance (Spearman distance, Ward linkage).

K	F1	AUC	NMI
5	0.814	0.884	0.613
10	0.816	0.890	0.649
15	0.823	0.893	0.659
20	0.828	0.897	0.635
30	0.833	0.901	0.670
50	0.816	0.888	0.659

Effect of distance metric.

Table 7 compares the three distance measures at $K = 15$. Kendall tau (F1 = 0.835) outperforms both Spearman (0.823) and Hamming (0.819). The two order-aware metrics (Kendall and Spearman) consistently beat the set-overlap metric (Hamming), confirming that the relative ordering of the top-K features and not just which features appear, carries a useful signal for the grouping participants.

Table 7. Effect of distance metric ($K = 15$, Ward linkage).

Metric	F1	AUC	NMI
Spearman	0.823	0.893	0.659
Kendall	**0.835**	**0.904**	0.658
Hamming	0.819	0.891	0.659

Effect of clustering threshold θ.

Table 8 shows that tighter thresholds (smaller θ, which cuts the dendrogram lower and produces more, smaller groups) yields higher F1 scores because each group is more internally consistent. Looser thresholds merge dissimilar participants together, reintroducing the cross-domain mixing problem that FedAvg suffers from.

 doi:10.20944/preprints202605.0501.v1

Table 8. Effect of distance threshold θ (Spearman distance, $K = 15$).

θ	F1	AUC	NMI
0.2	**0.827**	**0.899**	0.659
0.3	0.823	0.893	0.659
0.4	0.821	0.890	0.659
0.5	0.812	0.887	0.677
0.7	0.812	0.885	0.666
1.0	0.795	0.874	0.677

5.7. Error Analysis

Table 9 examines prediction-level changes for a representative small business participant, using a confusion matrix (a table showing counts of correct and incorrect predictions in each category).

Table 9. Confusion matrix comparison for a representative small business participant.

	LocalOnly		RankBridge	
	Pred. Benign	Pred. Phishing	Pred. Benign	Pred. Phishing
Actual benign	123	34	140	17
Actual phishing	38	105	38	105
F1	0.745		0.792	

RankBridge cuts the false-positive count in half (34 → 17), flagging far fewer legitimate emails as phishing, without missing any actual phishing. This improvement comes from averaging predictions across eight small-business participants that face the same threat profile; the combined model corrects for individual quirks that no single participant could catch on its own.

6. Discussion

6.1. Why Rank-Based Grouping Outperforms Weight-Based Grouping

The performance ordering in Table 4, RankBridge > LocalOnly > FedClust > FedAvg ≈ RandomCluster > IFCA, traces back to a single factor: how well each method handles mismatched feature spaces.

FedAvg averages model parameters directly across all participants. When one participant uses URL features and another uses email features, the averaged parameters carry corrupted information about both domains. IFCA avoids direct averaging through self-selection, but still exposes candidate models to incompatible updates in early rounds. FedClust groups by cosine similarity of weight vectors, which captures some broad structure but remains sensitive to feature-space mismatch because weight magnitudes are not comparable across different feature domains.

RankBridge groups by the one signal that remains interpretable regardless of how many features each participant uses: the relative order of feature importance. Two participants that both rank URL length, domain entropy, and suspicious TLD (top-level domain) at the top of their lists are correctly identified as similar even if their weight vectors occupy different spaces. Two participants from different domains will typically share few or no top-K features (since URL and email features occupy separate index ranges in the shared schema) and therefore receive a high pairwise distance, keeping them in separate groups.

6.2. Implications of Near-Perfect Grouping Accuracy

An NMI of 0.978 and ARI of 0.980 on real data are worth examining closely. Using only 60-byte ranked lists, with no gradients, no model weights, and no data statistics, the server correctly identified the sector affiliation of all 32 institutions across five organization types. Each participant's ranked list acts as a compact, privacy-preserving record of which threats matter most to them.

 doi:10.20944/preprints202605.0501.v1

This has real consequences beyond the accuracy numbers. A security operations center could use rank-based grouping to detect when a new attack campaign is disproportionately targeting a specific sector, to identify previously unrecognized structural similarities between organizations, or to flag when a single institution's threat profile shifts abruptly, potentially indicating a targeted attack or an internal compromise.

6.3. Privacy Analysis of the Ranked List Channel

RankBridge introduces one new information channel: the ranked list $R_i = [f_1, f_2, \ldots, f_K]$. An adversary who intercepts this list learns the identities and relative ordering of participant i's top-K features, but not their importance magnitudes, the model weights, any training samples, or the distribution of the training data. From the feature names alone, an adversary could infer the participant's domain (e.g., seeing URL-length feature indices implies URL data is in use), but this is qualitative guessing at best, not the pixel-level or token-level reconstruction that Zhu et al. [9] demonstrated is possible from gradients.

Even if an adversary assumes the simplest possible reconstruction (e.g., that importance scores decay uniformly with rank position), the ranked list does not carry enough numerical signal to recover the underlying importance magnitudes. The list reveals ordering, not scale. Standard privacy defenses already applied to model updates, such as secure aggregation and differential privacy, extend naturally to the ranked list channel as well.

6.4. Limitations

RankBridge has four practical limitations.

First, group representative selection defaults to the participant with the largest dataset rather than a quality-weighted approach. Smaller participants contribute to grouping but not to the representative model. Future work should explore weighting by local model accuracy alongside dataset size.

Second, building the distance matrix requires $O(N^2)$ pairwise comparisons, and Ward's linkage requires $O(N^2 \log N)$ time. At 32 participants both operations are nearly instantaneous; at thousands of participants, approximate nearest-neighbor methods would be needed to keep the server-side computation manageable.

Third, the clustering threshold θ must be set manually before deployment. Automatic selection, for example by finding the largest gap in the tree of merge distances, would remove this tuning step.

Fourth, SHAP values computed in early rounds are noisy because local models have not yet learned which features are reliably important. Figure 3 shows that grouping quality converges by round 10; a dedicated warm-up phase in which participants train locally before the first grouping step could speed up this convergence.

6.5. Broader Applicability

The mismatched feature-space problem addressed here is not unique to phishing detection. Any federated learning deployment where participating organizations use different types of data faces the same incompatibility. Industrial IoT networks combine sensors of different types; hospital networks use different imaging equipment; financial institutions track different sets of transaction features. In each case, grouping participants by which features their models find important, rather than by the numerical form of those models, is a practical path to collaboration that weight-based methods cannot offer.

7. Conclusions

We presented RankBridge, a federated learning protocol that groups heterogeneous participants by comparing ranked lists of SHAP feature importance rather than model weights or gradients. The reasoning is simple: weight-based aggregation breaks down when participants use different feature sets, a common situation in cross-institutional deployments, while feature importance rankings remain meaningful regardless of feature-space alignment.

 doi:10.20944/preprints202605.0501.v1

Experiments across 32 phishing detection participants in five organization types produced three consistent findings. First, parameter-averaging methods (FedAvg, IFCA) collapse below F1 = 0.28 on real heterogeneous data, confirming that mixing parameters across incompatible feature spaces is actively destructive. Second, RankBridge achieves F1 = 0.775 on real data and 0.854 on synthetic data, outperforming all five baselines including weight-based grouping (FedClust, F1 = 0.611) and working alone (LocalOnly, F1 = 0.744). Third, RankBridge recovers the correct organizational groupings with NMI = 0.978 and ARI = 0.980 while transmitting only 60 bytes per participant per round.

The organizations that benefit most are those with the weakest local detectors: healthcare institutions and small businesses that lack the data volume to train reliable models on their own. Rank-guided grouping connects them with same-sector peers whose models generalize to shared threat patterns, enabling meaningful federated collaboration without exposing private data, gradients, or model parameters. Taken together with prior single-institution work showing SHAP explanations are both accurate and user-accessible [12], these results suggest a broader trajectory: SHAP-based phishing detection can be made both explainable at the individual level and collaborative at the institutional level, with privacy intact throughout.

Future directions include adaptive grouping that tracks shifts in organizational threat profiles over time, formal differential-privacy guarantees for the ranked list channel, and approximate clustering algorithms that scale to deployments with hundreds or thousands of participants.

Author Contributions: Conceptualization, P.L.; Methodology, P.L.; Software, P.L.; Validation, P.L., R.Z. and T.L.; Formal Analysis, P.L.; Investigation, P.L.; Data Curation, P.L.; Writing – Original Draft Preparation, P.L.; Writing – Review and Editing, P.K., R.Z. and T.L.; Supervision, P.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ISCX-URL2016 dataset is publicly available from the Canadian Institute for Cybersecurity [20] (https://www.unb.ca/cic/datasets/url-2016.html). The Phishing Email dataset is publicly available on Kaggle [21] (https://www.kaggle.com/datasets/subhajournal/phishingemails). No new datasets were created as part of this work.

Acknowledgments: The authors thank the Department of Computer Science at the University of Texas Permian Basin for computational resources and the reviewers for their constructive feedback.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. *IEEE Access* **2022**, *10*, 65703–65727. https://doi.org/10.1109/ACCESS.2022.3183083.
2. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). PMLR, 2017, Vol. 54, *Proceedings of Machine Learning Research*, pp. 1273–1282.
3. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konecny, J.; Mazzocchi, S.; McMahan, H.B.; et al. Towards Federated Learning at Scale: System Design. *Proceedings of Machine Learning and Systems (MLSys)* **2019**, *1*, 374–388.
4. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning* **2021**, *14*, 1–210. https://doi.org/10.1561/2200000083.
5. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* **2020**, *37*, 50–60. https://doi.org/10.1109/MSP.2020.2975749.

 doi:10.20944/preprints202605.0501.v1

6. Hsu, T.M.H.; Qi, H.; Brown, M. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *arXiv preprint arXiv:1909.06335* **2019**.
7. Ghosh, A.; Chung, J.; Yin, D.; Ramchandran, K. An Efficient Framework for Clustered Federated Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020, Vol. 33, pp. 19586–19597.
8. Sattler, F.; Müller, K.R.; Samek, W. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *32*, 3710–3722. https://doi.org/10.1109/TNNLS.2020.3015958.
9. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2019, Vol. 32, pp. 14774–14784.
10. Lyu, L.; Yu, H.; Yang, Q. Threats to Federated Learning: A Survey, 2020, [arXiv:cs.CR/2003.02133].
11. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017, Vol. 30, pp. 4765–4774.
12. Lim, P.; Huerta, R.; Sotelo, A.; Quintela, A.; Husák, M.; Kumar, P. VeriPhish: Bridging AI Explainability and Accuracy in Phishing Detection Through XAI and LLMs. In Proceedings of the 2025 IEEE International Carnahan Conference on Security Technology (ICCST), 2025, pp. 1–6. https://doi.org/10.1109/ICCST63435.2025.11295116.
13. Lim, B.; Huerta, R.; Sotelo, A.; Quintela, A.; Kumar, P. EXPLICATE: Enhancing Phishing Detection through Explainable AI and LLM-Powered Interpretability, 2025, [arXiv:cs.CR/2503.20796]. https://doi.org/10.48550/arXiv.2503.20796.
14. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. In Proceedings of the Proceedings of Machine Learning and Systems (MLSys), 2020, pp. 429–450.
15. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nature Machine Intelligence* **2020**, *2*, 56–67. https://doi.org/10.1038/s42256-019-0138-9.
16. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017, Vol. 30, pp. 3146–3154.
17. Spearman, C. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* **1904**, *15*, 72–101. https://doi.org/10.2307/1412159.
18. Kendall, M.G. A New Measure of Rank Correlation. *Biometrika* **1938**, *30*, 81–93. https://doi.org/10.1093/biomet/30.1-2.81.
19. Ward, J.H. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* **1963**, *58*, 236–244. https://doi.org/10.2307/2282967.
20. Mamun, M.S.I.; Rathore, M.A.; Lashkari, A.H.; Stakhanova, N.; Ghorbani, A.A. Detecting Malicious URLs Using Lexical Analysis. In Proceedings of the Proceedings of the 10th International Conference on Network and System Security (NSS 2016). Springer, 2016, Vol. 9955, *Lecture Notes in Computer Science*, pp. 467–482. https://doi.org/10.1007/978-3-319-46298-1_30.
21. Subhajournal. Phishing Email Dataset. https://www.kaggle.com/datasets/subhajournal/phishingemails, 2023. Kaggle.