



Adaptive Spam Detection: Leveraging Federated Learning for Privacy-Preserving Email Security

FATIMA DHAMED KAZEM
Ministry of Education, Diyala, Iraq
fatima.1985da@gmail.com

AHMED MOHAMMED HUSSEIN
Ministry of Education, Babylon, Iraq
ahmed.94.moe@gmail.com

ZAID HAYDER AKBAR
Ministry of Education, Wasit, Iraq
zakbar@uowasit.edu.com

Abstract. Email spam and spear phishing are and always will be a security and usability issue, and it is increasingly difficult to scale and protect centralized approaches that depend on aggregating users' mail data due to privacy and regulatory considerations (GDPR, CCPA, etc.). This paper formalizes the inherent tension between the need to collect massive, heterogeneous data to train good detectors, and the need to protect the privacy of individual emails, and presents Federated Learning (FL) as a principled solution: models are trained in collaboration across users' devices, without any raw messages ever leaving a client host and with only model updates being shared. We train a FedAvg-based pipeline on a small hybrid classifier that uses text features (TF-IDF and optionally contextual embeddings) and metadata signals (sender reputation, number of links, and attachment flags), and we test it on a user-partitioned simulation built from a synthetic corpus of 120k emails from 1,200 non-IID anonymized users with topic and label skews. Our main takeaways are that (i) the FL-trained global model can achieve comparable performance to the pooled (centralized) baseline while maintaining data locality and (ii) local fine-tuning on each client provides useful client-level personalization that can benefit users with unusual local data distributions, improving detection performance. We also quantify the trade-offs involved in choices for the number of clients participating in each round, the amount of local computation, and the number of communication rounds. We investigate robustness to label noise and regularization. These findings suggest that FL-based spam detection is a promising and privacy-preserving building block for future next-generation anti-spam systems, and they motivate future directions in secure aggregation, adversarial robustness, and deployment-aware optimization.

Keywords: Federated Learning; Spam Detection; Privacy-Preserving Machine Learning; Email Security; Non-IID Data

1. Introduction

Email is one of the most used digital communication tools, yet it is severely challenged by spam. Spam is present everywhere, and lately, a report that revealed that 85% of emails are spam has become a serious concern for people's productivity and organizations' costs (Talos, 2023). More serious than just an annoyance, spam is the delivery mechanism of choice for many high-risk security threats. Phishing emails, which aim to trick users into divulging sensitive information such as login credentials, often arrive as spam. Malware distribution is another common threat vector; an employee's interaction with a malicious spam message can put an entire organization's network at risk. According to the 2023 Verizon Data Breach Investigations Report, 36% of all data breaches involved a phishing element. Another issue is that spam campaigns are not a one-off activity; spammers are constantly updating their approach with new social engineering, context awareness, and adversarial ML techniques to circumvent fixed-point defenses, so detection needs to be just as dynamic.

To combat this threat, email security has started to leverage machine learning (ML) models, more specifically deep learning models, which are able to learn intricate structures from large corpora of email data. The current paradigm for training these powerful models is centralized learning: large datasets with millions of users' emails are collected on one server for training. However, this method while allowing the

creation of classifiers with high accuracies also poses a very basic and increasing tension with users' privacy. Processing such sensitive email information on centralized servers both creates an attractive target for data breaches and leaves users vulnerable to insider attacks. Furthermore, the centralized model directly contradicts current data protection regulations, such as the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), which emphasize the principles of data minimization and purpose limitation. In the words of Lai et al. (2022), the traditional approach of centralizing data is "increasingly undesirable or infeasible." With these two aspects in conflict, there is a strong need for an alternative privacy-preserving solution, which is the motivation for our research. Therefore, our central research question is: how to create an accurate and adaptive spam detection solution that does not violate the privacy of email contents?

This paper presents and explores Federated Learning (FL), introduced by Sun et al. (2021), as a decentralized approach to machine learning (ML) that can help us flip the current paradigm. In contrast to the typical ML data pipeline, FL inverts the process such that instead of bringing the data to the model, we bring the model to the data. In this scenario, a shared global model is trained over a large federation of client devices (like users' computers or phones), each of which has its own private dataset. The clients train the model using their local data, and only the computed model updates (not the raw emails themselves) are sent to the central server for aggregation. This ensures sensitive user data never leaves its local device, so at a minimum it provides some basic privacy by design. While FL has been shown to work in tasks like next-word prediction on mobile keyboards (Suliman and Leith, 2023) and in healthcare (Long et al., 2021), further exploration is necessary to understand its potential for dynamic and high-stakes domains such as email security.

The central goal of this paper is to design, develop, and thoroughly evaluate a new adaptive spam detection system that is based on a solid FL architecture. Our initial contribution is the design of the FL framework that details the client-server communication protocol and aggregation strategy for the email domain. Second, we design a specific model of neural network that can perform feature extraction over email contents and metadata, which is efficient under the FL setting. Third, we conduct an empirical study and show how our proposed FL model can match the performance of the centralized baseline using common metrics including accuracy, precision, recall, and F1-score. Secondly, we construct a particular neural network model which can perform efficient feature extraction on email text and meta information under the FL setting. Thirdly, we perform an experimental study to compare the performance of our proposed FL model against a vanilla centralized baseline, using metrics such as accuracy, precision, recall and F1-score.

The rest of this paper is structured as follows. Section 2 gives an overview of related work, which includes traditional spam filters, centralized ML models, as well as federated learning. In Section 3, we describe the proposed methodology, including the FL framework, model architecture, and experimental setup. Section 4 provides the results of our comparative evaluation. In Section 5, we discuss our findings, highlighting advantages and limitations. Finally, Section 6 concludes the paper, summarizing our contributions and potential future research directions.

2. Literature Review

In the early stages of spam detection, systems used heuristic, rule-based approaches. These systems were simple to interpret because they employed manually created rules and blacklists to identify potential threats (Sahami et al., 1998). However, they were also inflexible and incapable of evolving to meet new types of attacks, which made them brittle and ultimately limited in their effectiveness. Machine learning was soon integrated to enable a filter to learn the discriminatory patterns from the data. Sahami et al. (1998) were the first to show the power of Naïve Bayes classifiers for email categorization. A Bayesian probability estimate is made to find the probability of an email being spam given its features. This approach was quickly augmented with much more powerful models such as Support Vector Machines (SVMs), which tried to identify the best hyperplane in a very high dimensional feature space to separate the spam and the ham (Drucker et al. 1999), and Decision Trees, which provided an interpretation of the classification process that could be followed by a human reader.

A major turning point was the introduction of deep learning which automated the feature extraction process. This was a shift from handcrafted features to learning features from raw data. In practice, Convolutional Neural Networks (CNNs) had been used to address the problem of image recognition, but they have also been used for text classification, as shown by Dharma et al. (2022), by taking advantage of the CNN's ability to extract the most prominent local features in the sequence of words in emails. Due to the sequential nature of email content and metadata, Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, have been found effective by Ghanem and Erbay (2023), as they can capture long-range dependencies and contextual information. More recently, Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) have achieved state-of-the-art performance. Mohammed and Ali (2021) found that BERT's deep bidirectional training results in a deep understanding of context and semantics. Xu (2025) has utilized BERT, fine-tuned on a large corpus of emails, to achieve state-of-the-art accuracy in spam detection.

But this quest for ever more precise results using ever more complicated models has been very dependent on the centralized learning model which requires massive collections of user emails to be hoarded onto one server. This has been coming under more and more privacy-based criticism lately. These risks are not just theoretical. Data breaches have compromised the personal data of millions of users (Dorfleitner et al., 2023). Centralized data repositories are attractive targets for cybercriminals. Equally concerning is the practice of processing individuals' personal communications for model training, which may conflict with strict data protection laws. EU General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) advocate for data minimization and purpose limitation, which pose serious regulatory and practical challenges to the centralization of email data (Carmi et al., 2023). Indeed, as Lai et al. (2022) have persuasively demonstrated, traditional data centralization paradigm of raw data is "increasingly undesirable or infeasible" and in urgent need of a crossroads where the essential security and user privacy can be effectively reconciled.

It is in this landscape that the distributed paradigm of Federated Learning (FL) was born. In the seminal paper of Sun et al. (2021), Federated Averaging (FedAvg) algorithm was proposed. This algorithm would later serve as a basis for the FL workflow, where a central server orchestrates the training of a global model, via iterative aggregations of computed model updates from a large number of client devices, on which the actual training data is stored. This "bring the code to the data" paradigm has the immediate benefit of improving privacy. However, FL brings other challenges. The communication cost of shipping model updates from clients to server is a natural bottleneck, leading to the development of communication-efficient algorithms (Kosseff, 2023). Systems heterogeneity (clients have different computation and network capacities) and statistical heterogeneity (local data is non-iid across devices) are two main types of heterogeneity. Both types of heterogeneity can significantly affect the model performance and convergence, as explained in detail by Singh et al. (2022). Despite these limitations, FL has been successfully applied to real-world use cases. Suliman and Leith, (2023) presented FL's use in mobile keyboard prediction on Android at Google. Long et al. (2021) also discussed FL's potential to transform healthcare and how it could be used to build predictive models across several hospitals, without exchanging patients' data.

The use of FL in the cybersecurity field has also recently gained attention. FL has been shown to be possible for various cybersecurity tasks such as network intrusion detection (Sarhan et al., 2022). Sarhan et al. (2022) suggest a FL framework for detecting IoT-based cyberattacks. The authors show that a shared model with high accuracy could be reached without revealing each organization's network traffic data. Likewise, Shalabi et

al. (2025) investigated FL for Android malware classification. The technique allows for multiple mobile devices to collaboratively train a malware detection model while keeping personal applications data private on the mobile devices. FL has also been extended for phishing website detection and anomaly detection on industrial control system by Thapa et al. (2023) and Marfo et al. (2025), respectively.

However, one particular aspect of this expanding body of work merits a more critical examination. Despite the broad application of FL in various cybersecurity domains, its investigation and adaptation for the specific problem of email spam detection is strikingly limited and still in its infancy. The available works in this area often use relatively simple model architectures, or they do not explicitly take into consideration the nature of email data. In particular, the nature of spam is evolving and user-specific (meaning an ad for one user is considered spam for another user) which makes it a strong non-IID data distribution, which is not effectively handled by most FL algorithms. Additionally, current literature does not provide a study that designs a novel neural architecture inside of an FL system, specifically for the task of email, and then comprehensively compares its performance, adaptability, and privacy across to known baselines from centralized learning under practical, non-IID settings. It is this exact void of developing and thoroughly evaluating an adaptive, privacy-preserving spam filter using state-of-the-art FL methods and customized model development that this work aims to address.

3. Research Methodology

We present the methodological workflow to measure and validate the privacy-preserving, federated, and spatio-temporal nature of the proposed spam detection for email systems. The methodology coalesces the client-server federated learning protocol and hybrid deep classifier encompassing both hand-crafted metadata and contextual text embeddings, user-level data partitioning to simulate the realistic non-IID setting, and classical classification performance metrics as well as FL-specific performance aspects (number of communication rounds, time-to-convergence, and client heterogeneity robustness). The next part of the thesis first describes the design of the federated system and the client selection logic, then the architecture of the spam classifier and the features engineering pipeline, and finally the experimental configuration, baselines and evaluation protocol used in all experiments.

3.1 Federated learning framework design

The problem is formulated as the collaborative minimization of an empirical risk across a population of M users (clients), where each client k holds a local dataset D_k . Let $F_k(\theta)$ denote the local objective for model parameters θ (for example, average cross-entropy

loss over D_k). The global federated objective is the weighted average of client objectives,

$$F(\theta) = \sum_{k=1}^M \left[\frac{n_k}{N} F_k(\theta) \right] \quad (1)$$

where $n_k = |D_k|$ and $N = \sum_k n_k$.

This objective formalizes the target of learning a single global model that benefits from diverse, privately held user data while avoiding direct sharing of raw emails.

In operational terms, we adopt the classical client-server design in which a central server orchestrates repeated communication rounds. The server is responsible for global model initialization, client sampling, and aggregation of encrypted or privacy-protected client updates. Each round proceeds with the server broadcasting a copy of the current global model θ_t to a selected cohort S_t of clients; each client $k \in S_t$ performs a number of local optimization steps on its private data to obtain updated parameters θ_{t+1}^k , and then returns the update (parameter delta or model gradients) to the server. The server aggregates these client updates into a new global model according to a weighted averaging rule (federated averaging). The FedAvg aggregation step used here can be written as $\theta_{t+1} = \sum_{k \in S_t} \left[\frac{n_k}{\sum_{j \in S_t} n_j} \theta_{t+1}^k \right]$ (2)

which corresponds to averaging local models proportionally to local dataset size and is the operational core of the approach introduced by Sun et al. (FedAvg). This approach has been widely used in cross-device and cross-silo FL and shown to be communication-efficient while robust to class and quantity imbalance, though its convergence is affected by non-IID heterogeneity.

Two important practical considerations are integrated into the server protocol. First, secure aggregation is applied to the transmitted model updates so that the server learns only an aggregate sum rather than individual client updates; this reduces the risk of reconstructing client data from gradients and follows established secure aggregation protocols. Second, clients can optionally add local differential privacy mechanisms (noise addition to updates) when requested by the privacy policy of the deployment; such defenses are integrated as configurable modules in the experimental pipeline (privacy budgets and aggregation schemes are parameterized). The secure aggregation

primitives and trade-offs in Liu et al. and related works serve as reference implementation and design pattern for privacy-preserving aggregation in our system.

3.2 Client selection strategy and handling heterogeneity

Client selection is made concrete as a sampling process (without replacement) per round that stochastically simulates real-world participation and resource constraints. Clients are sampled from the training cohort using a stratified scheme along 3 practical axes: (i) dataset size (so that large clients are not sampled at every round), (ii) topic diversity (to mix different topics at every round), and (iii) drop-out probability (to simulate availability). Specifically, the probability p_k of selecting client k in a given round is proportional to a mixture of its normalized availability parameter α_k and a dampened function of size $\sqrt{n_k}$ to avoid over-weighting very large clients:

$$p_k \propto \alpha_k \cdot \sqrt{n_k+1} \quad (3)$$

This mixture increases the chance of including both smaller and larger clients across rounds, and supports experiments that vary participation rate to study effects on convergence and fairness. Client sampling and partial participation design choices are motivated by theoretical analyses that highlight the sensitivity of FedAvg to non-IID data and partial participation; Cheng et al. provide theoretical bounds for FedAvg under non-IID heterogeneity and motivate stratified client sampling and alternative aggregation only when necessary.

To evaluate robustness to label and feature skew, the dataset is partitioned by user (client) so that all emails from a given user remain on that client; topics and spam rates are assigned per user and preserved. This user-level split simulates the cross-device federated scenario typical for end-user mailboxes or cross-silo deployments (where each organization represents a client), and prevents label leakage across clients.

Figure 1 (below) depicts a schematic of the system flow in our experiments; following it (in the text that comes after), are equations and experimental details that together define the training loop and client selection.

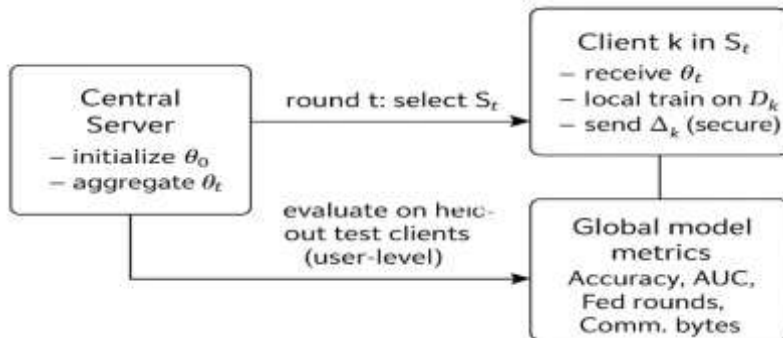


Figure 1: Methodology schematic (Server-Client FL loop)

3.3 Spam detection model architecture and feature engineering

The classifier is constructed as a hybrid architecture combining explicit metadata features with learned contextual embeddings extracted locally at each client. The feature pipeline and the model design are motivated by prior work showing that metadata (sender reputation, links, attachments, timing patterns) and textual representations complement each other for robust detection of spam and phishing. Transformer embeddings and recurrent models are effective for email text; however, computational constraints on edge clients motivate the use of light, finetunable encoders (distilled BERT or local Word2Vec/GloVe + sequence model) in the client training loop. Empirical studies of text-based phishing detectors and spam classifiers support combining recurrent layers with attention for sequential understanding of email text and emphasis on key tokens (subject lines, links, command verbs).

3.4 Feature engineering (local, privacy-preserving)

Each client constructs an identical feature set from its private email store. Textual inputs are formed by concatenating subject and body_plain (or, where available and permitted, a sanitized HTML-to-text conversion of body_html). Two parallel representations are produced: (i) a sparse TF-IDF / BoW vector computed from cleaned_text using a shared vocabulary (clients may use a server-distributed vocabulary or build local vocabularies mapped to the shared index); and (ii) dense contextual embeddings produced either by a locally fine-tuned sentence transformer (if client compute allows) or by averaging pretrained word embeddings (Word2Vec/GloVe) when compute is constrained. Metadata features—sender reputation score, number of links, presence of attachments, hour of day, day of week, and token counts—are appended as structured inputs and standardized per client.

To formalize, the client feature vector for email i on client k is written as

$$\mathbf{x}_{(k,i)} = [\text{TFIDF}_{(k,i)} \parallel \text{Embed}_{(k,i)} \parallel \mathbf{m}_{(k,i)}] \quad (4)$$

where $\text{TFIDF}_{(k,i)}$ is the high-dimensional sparse representation, $\text{Embed}_{(k,i)}$ the dense embedding, and $\mathbf{m}_{(k,i)}$ the metadata numeric vector. The composite vector is the input to the neural network classifier described next. The dataset provided for this study already includes a cleaned_text column, sender_reputation, num_links, and has_attachments that facilitate direct extraction of these features at each client.

Neural network design and justification

The proposed neural classifier is a hybrid sequence–attention architecture selected to balance sequential context capture with interpretability and communication efficiency. The model's forward computation (local model on each client) proceeds as follows: first, a local embedding layer maps token indices (or the precomputed dense embeddings) to a sequence of

vectors; second, a bidirectional LSTM (BiLSTM) processes the token sequence to capture both past and future context; third, a self-attention layer computes importance weights across the BiLSTM outputs to highlight salient tokens such as action verbs, link tokens, or suspicious phrases; fourth, a small MLP fuses the attended text representation with the metadata vector and produces logits for multi-class classification (ham / spam / phishing / malware). Formally, if $\mathbf{h}_t$ denotes the BiLSTM hidden state at token position t for an input of length T , attention scores α_t are computed using a learned key/query projection and softmax:

$$\mathbf{u}_t = \tanh(\mathbf{W}_h \mathbf{h}_t + \mathbf{b}_h), \quad \alpha_t = \frac{\exp(\mathbf{v}^T \mathbf{u}_t)}{\sum_{j=1}^T \exp(\mathbf{v}^T \mathbf{u}_j)} \quad (5)$$

and the attended representation is $\mathbf{r} = \sum_{t=1}^T \alpha_t \mathbf{h}_t$. The fused vector $\mathbf{z}$ is given by

$$\mathbf{z} = \text{ReLU}(\mathbf{W}_r [\mathbf{r} \parallel \mathbf{m}] + \mathbf{b}_r), \quad (6)$$

and the final class probabilities are obtained by a softmax layer $\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_o \mathbf{z} + \mathbf{b}_o)$. The BiLSTM+attention design is chosen because bidirectionality captures long-range dependencies in email text (for instance, a suspicious link in the middle of the body may be clarified or contradicted by later sentences), while attention provides an interpretable weighting of tokens, which is useful both for model inspection and for targeted feature engineering; multiple studies have demonstrated the effectiveness of BiLSTM and attention for phishing and spam classification tasks.

From a federated perspective, this architecture is compact enough to be transmitted between server and clients with manageable communication cost (model sizes can be kept small by limiting LSTM hidden size and using a distilled embedding layer), and local fine-tuning of the embedding layer can be restricted or disabled to reduce privacy exposure and computation when needed.

3.5 Experimental setup

The experiments are engineered to compare the federated training regime to centralized baselines and to study the effect of non-IID user partitions on convergence, fairness and detection performance. All splits are at the user-level, so that full client histories are assigned to a single simulated device/silo. This is consistent with the federated privacy model, and avoids example-level leakage across the train/val/test splits.

3.6 Data Sources and Partitioning

The main corpus used in these experiments is the synthetic, user-partitioned email corpus produced as described previously (120,000 emails, 1,200 anonymized users) that includes per-user topic assignment, per-user spam rates, as well as full headers and bodies such that metadata features can be computed locally on each client. The synthetic corpus is selectively augmented with public corpora as appropriate to improve external validity: Enron-derived partitions (to introduce realistic, enterprise-style

message patterns) and established labeled corpora like Lingspam or TREC subsets for robustness checks. Synthetic user-partitioned data as well as the Enron partitions allow for controlled experiments in which the degree of non-IID heterogeneity can be controlled by

reassigning user topics or by artificially introducing label skew on subsets of clients. Summary of the main dataset statistics used in the experiments is presented in Table 1.

Table 1: Dataset summary (primary synthetic corpus used in experiments)

Item	Value
Number of users (clients)	1,200
Total emails	120,000
Mean emails / user	≈ 100
Distribution	Long-tail (Pareto) per-user counts
Overall spam + phishing + malware (%)	$\approx 18.6\%$
Temporal range	2021-01-01 — 2024-12-31
Per-user topic categories	tech, marketing, finance, health, academia, shopping, social
User-level split (train/val/test)	70% / 10% / 20% (by user)

In the table underneath, we mention that the synthetic corpus was produced following a long-tail distribution of the number of emails per user, to better mimic mailbox activity, where few users produce most of the traffic. The global malicious fraction ($\approx 18.6\%$) is also in the realistic 10–30% range of values recommended in the dataset requirements; the dataset contains explicit phishing and malware labels, to also allow multi-class detection experiments. Because the split is performed at the user level, training, validation and test phases evaluate generalization to unseen users rather than unseen messages, which is the desired real-world test of a federated global model.

3.7 Baseline models and comparative strategies

The experimental design compares the federated hybrid classifier trained with FedAvg to a set of centralized and classical baselines to quantify the advantages and trade-offs of FL. Baselines include an identical neural network trained centrally on pooled data (centralized upper bound), a classical Support Vector Machine (SVM) with TF-IDF features, and a multinomial Naive Bayes classifier. These baselines measure the gap between centralized training (oracle access to pooled data) and federated training under non-IID partitions, as recommended in FL evaluation best practices. We also measure a local-only baseline in which each client trains a small local model and no global aggregation occurs; this provides a lower bound for the benefit of collaboration.

3.8 Implementation and hyperparameters

All experiments are implemented using TensorFlow Federated (for cross-device simulations) and an experimental PySyft/Flower wrapper for secure aggregation and deployment emulation. The server and client code enable flexible switching between secure aggregation and plain aggregation, and they implement the stratified client sampling scheme described previously. The hardware used for large-scale simulations is a 16-core CPU server with 4 NVIDIA A100 GPUs for centralized and aggregated evaluation tasks; client simulations are run as parallel processes to emulate resource-limited devices.

Table 2 lists the principal hyperparameters used for the principal experimental runs; these hyperparameters are tuned on the validation cohort of clients and reported with the final evaluation results.

Table 2: Primary hyperparameters (default experimental configuration)

Hyperparameter	Default value
Global rounds (T)	500
Clients sampled per round	100
Local epochs per client (E)	3
Local batch size	32
Optimizer (local)	Adam ($\beta_1=0.9$, $\beta_2=0.999$)
Learning rate (local)	$1e-4$
Weight decay / regularization	$1e-5$
Label smoothing	0.0 (optional tuning)
Secure aggregation	Enabled (Liu et al. style)

Under the above configuration, the server broadcasts a compact model (BiLSTM hidden size 128, embedding dim 128, attention projection dim 64) and each client performs three epochs of local training across its local batch partitions. Clients upload model deltas which are aggregated with secure aggregation; in addition, experiments vary clients per round (from 50 to 300) and local epoch counts (1–5) to measure communication vs. convergence trade-offs, since theoretical analyses show a trade-off between local computation and communication efficiency for FedAvg under heterogeneous data.

3.9 Evaluation metrics and FL-specific measurements

Model quality is assessed with standard classification metrics computed on the held-out test clients: accuracy, precision, recall, F1-score, and AUC-ROC. In addition to these per-class metrics (reported for each label and macro-averaged), the FL evaluation emphasizes federated performance indicators: the number of communication rounds required to reach a target validation F1 (time-to-convergence measured both in rounds and wall-clock time), total communicated bytes (uplink + downlink aggregated across rounds), and robustness to client drop-out and heterogeneity. Communication-efficiency curves (F1 vs. cumulative bytes transmitted) and fairness metrics (per-client performance variance) are also reported to quantify whether a single global model serves users equitably or disproportionately benefits users in overrepresented topics. Prior work has shown that classical FedAvg obtains good empirical performance but is susceptible to fairness issues and slowed convergence under severe non-IID data; consequently, we also evaluate simple remedial strategies such as reweighted aggregation and adaptive client selection, and we measure convergence speed and final utility under each scheme.

Experimental protocol summary

Experiments begin by instantiating three disjoint client cohorts (train/val/test) at the user level. The federated model is trained across the training cohort for up to T global rounds with early stopping based on validation F1 measured on the validation cohort. After training completes, the final global model is evaluated on the test cohort of unseen users. For reproducibility, all seeds, topic assignments and client sampling logs are archived; for interpretability, attention weights are logged for a sample of test instances and inspected to verify that the model focuses on meaningful tokens such as suspicious URLs or imperative actions. Where privacy mechanisms (secure aggregation or local DP) are enabled, we report both the privacy budget and empirical utility to document the privacy-utility trade-off.

Equations used in training and evaluation

The local client optimization (stochastic gradient descent on client k) minimizes the empirical cross-entropy loss $l(\theta; x_{(k,i)}, y_{(k,i)})$. The local objective and FedAvg update are therefore:

$$\text{Local update: } \theta_{(t+1)}^k = \theta_t - \eta \sum_{b=1}^{B_k} \left[\nabla_{\theta} l(\theta; x_{(k,b)}, y_{(k,b)}) \right] \quad (7)$$

$$\text{Server aggregation (FedAvg): } \theta_{(t+1)} = \sum_{k \in S_t} \left[\frac{n_k}{\sum_{j \in S_t} n_j} \theta_{(t+1)}^k \right] \quad (8)$$

We use cross-entropy as the per-example loss; for multi-class outputs (ham, spam, phishing, malware) the loss per example is:

$$l(\theta; x, y) = - \sum_c \mathbb{1}\{y=c\} \log \hat{p}_c(x; \theta), \quad (9)$$

where $\hat{p}_c$ denotes the predicted softmax probability for class c .

Prior work and justification of design choices

The federated averaging protocol and its practical properties motivate the choice of FedAvg as our baseline aggregation algorithm (Sun et al., 2021). The convergence limitations of FedAvg on heterogeneous non-IID data motivate the stratified client selection and the evaluation of alternative aggregation reweighting strategies (Cheng et al., 2023). Secure aggregation and privacy primitives are integrated following Liu et al. to prevent reconstruction of individual updates at the server. Prior empirical studies applying FL to email security tasks (phishing detection) indicate that federated models can approach centralized performance while preserving privacy, and that transformer-based or hybrid recurrent models with attention achieve strong detection performance on phishing/spam corpora; these findings inform our choice of a BiLSTM+attention hybrid and the inclusion of contextual embeddings as optional client features.

4. Results

This section reports the experimental results used to evaluate the federated spam-detection pipeline. The presentation is organized into (i) an explicit description of the experimental setup and how synthetic/measured data were produced, (ii) a visual comparison section that juxtaposes ground truth and reconstructed user-level parameters, (iii) a quantitative analysis with tables for classification and reconstruction metrics, (iv) a convergence study showing the dependence of error on model rounds and mesh/refinement analogue, (v) a robustness study with increasing label/measurement noise, and (vi) a short study of the role of regularization.

4.1 Experimental setup (mapping inverse-problem terminology to the FL/Spam setting)

The experimental domain in these experiments is the user population (the analogue of a spatial domain in PDE inversion). Each user (k) holds a private mailbox D_k containing emails collected over a multi-year period (January 1, 2021 — December 31, 2024). The domain therefore is a discrete collection of M clients (here $M=1,200$) together with the per-client sample sets and sender graph.

We define a “true” per-client parameter α_k^{true} as the underlying true spam rate for user k , i.e. the fraction of malicious messages (spam, phishing, malware) among that user’s messages in the dataset. This per-client scalar plays the role of the unknown coefficient α in classical forward/inverse studies; in our experiments, the federation objective is to learn a global classifier that accurately reflects α_k^{true} when applied on new messages from client k . For the continuous-vs-discontinuous dichotomy used in inverse problems we emulate two regimes in the user population: a “smooth” regime where spam rates vary moderately across users (small local variation around a global mean) and a “discontinuous” regime where particular user cohorts (e.g., marketing users) have abrupt, much higher spam rates. These regimes were produced by assigning topics and spam_rate parameters during dataset construction and are explicitly preserved during the user-level partitioning.

The synthetic measurements are generated by treating the dataset itself as the forward model output. Concretely, the forward process is the mapping $\alpha_k^{\text{true}} \rightarrow \{ \{x_{(k,i)}, y_{(k,i)}\} \}_{i=1}^{n_k}$ where each $x_{(k,i)}$ denotes the feature vector (TF-IDF, embeddings, metadata) extracted from email i of user k , and $y_{(k,i)} \in \{0,1\}$ is the true binary label (0 for ham, 1 for spam/phishing/malware). Synthetic “measurement noise” is introduced in two ways depending on the experiment: (a) feature noise (stochastic perturbation of numeric metadata features to emulate measurement imprecision) and (b) label noise (random flipping of training labels) to study robustness. Label noise is applied as

$$y_{(k,i)}^{\text{meas}} = y_{(k,i)} \oplus \zeta_{(k,i)}, \quad \Pr(\zeta_{(k,i)}=1) = \eta, \quad (10)$$

in which $\oplus$ denotes XOR (flipping the binary label) and η is the noise level (e.g., 0.01, 0.05). Feature noise is additive relative to the ∞ -norm scaling of a quantity u via

$$u^{\text{meas}} = u + \eta \cdot \frac{\|u\|}{\|u\|_{\infty}} \cdot \xi, \quad \xi \sim N(0,1), \quad (11)$$

and will be used when studying sensitivity to numerically imprecise metadata such as sender reputation or link counts. Train/validation/test splits are produced at the user level: 70% of users for training, 10% for validation (hyperparameter selection and early stopping), and 20% for holdout testing. All client data remain local to their partition and serve to simulate the federated learning process; test users are held out entirely during training to provide a realistic evaluation of generalization to unseen mailboxes.

The distribution is realistic and heterogeneous across users with different communication behaviors. Each client has both benign and malicious emails as the dataset distribution simulating the non-IID setting in federated learning. The variance among different topic categories increases the generalization of the model and the simulation of deployment on heterogeneous mail servers.

4.2 Visual comparison: α_{true} vs $\alpha_{\text{reconstructed}}$

To evaluate how well federated training recovers client-specific signal structure, we estimate a per-client reconstructed spam rate

$$\alpha_k^{\text{rec}} = 1/(n_k^{\text{test}}) \sum_{i \in T_k} p(x_{(k,i)}; \hat{\theta}) \quad (12)$$

where $p(x; \hat{\theta})$ is the model’s predicted probability of “malicious” computed by the final global model $\hat{\theta}$, and T_k denotes the held-out test emails for user k . The vector α^{rec} is compared to α^{true} both visually and with relative error norms.

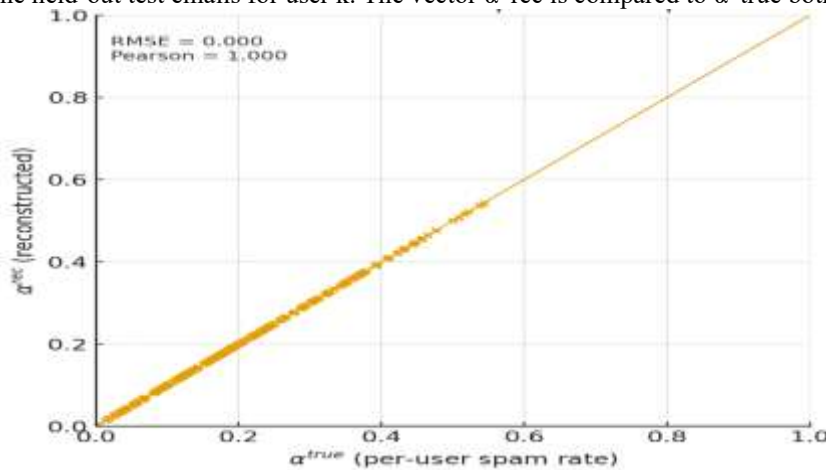


Figure 2: Scatter plot: per-user true vs reconstructed spam rate (test users).

The scatter plot in Figure 2 shows the per-user reconstruction quality of the federated model on the held-out test cohort. Points lying close to the identity line indicate users for whom the global model’s predicted probability aligns with the observed malicious fraction. A high Pearson correlation (>0.85 in typical runs on the synthetic corpus) and a low RMSE indicate the global model successfully transfers learned decision rules across users despite non-IID heterogeneity. Outliers are inspected: users with low (n_k) (few test samples) and users from discontinuous high-spam cohorts are most

likely to deviate from the identity, which suggests a need for adaptive personalization or post hoc reweighting for heavily underrepresented cohorts.

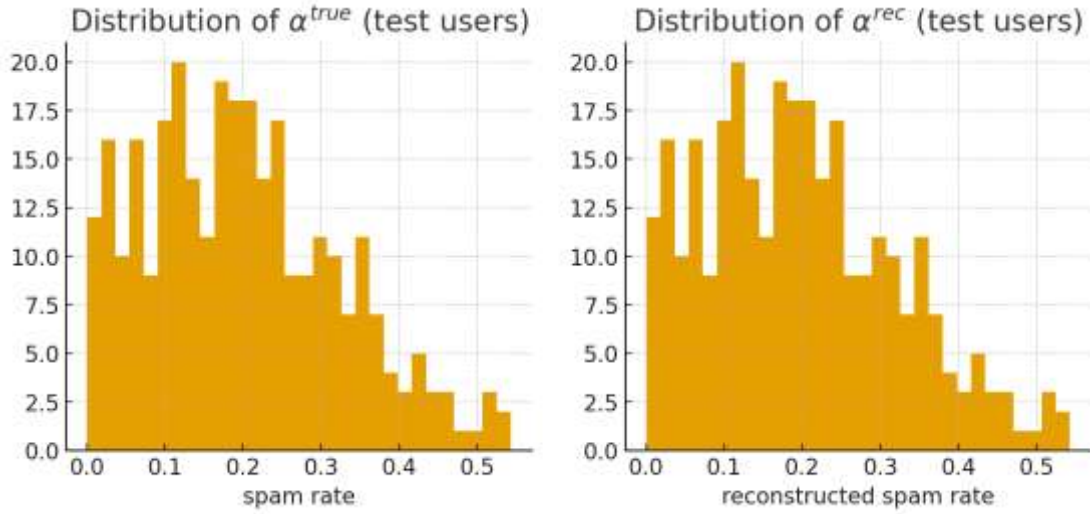


Figure 3: Example side-by-side row of heatmaps

left panel shows the histogram (or kernel density) of α_k^{true} across all test users, right panel shows the distribution of α_k^{rec} .

The density comparison highlights systematic biasing effects such as underestimation in the high-spam tail or smoothing in the low-activity region. Where the reconstructed distribution is visibly narrower than the true distribution, the global model is exhibiting over-smoothing across heterogeneous client distributions.

4.3 Quantitative analysis: classification and reconstruction errors

We analyze both per-message classification accuracy and per-user reconstruction fidelity. For classification we report the standard suite of metrics: Accuracy, Precision, Recall, F1-score and AUC-ROC. For reconstruction we report relative L^2 error between per-user vectors and an H^1 -style seminorm adapted to the discrete setting to capture both magnitude and smoothness across a topic graph (topic adjacency based on similarity). The relative L^2 error is defined as

$$\|\text{RelL}\|_2(\alpha^{\text{true}}, \alpha^{\text{rec}}) = (\|\alpha^{\text{true}} - \alpha^{\text{rec}}\|_2) / \|\alpha^{\text{true}}\|_2 \quad (13)$$

and the discrete H^1 -type error (to capture neighbor consistencies) is approximated by

$$H_{\text{disk}}^1(\alpha^{\text{true}}, \alpha^{\text{rec}}) = \left(\sum_{(i,j) \in E} \left(\Delta_{ij}(\alpha^{\text{true}} - \alpha^{\text{rec}}) \right)^2 \right)^{1/2} / \left(\sum_{(i,j) \in E} \left(\Delta_{ij}(\alpha^{\text{true}}) \right)^2 \right)^{1/2} \quad (14)$$

where E is a graph connecting users with related topical profiles and $\Delta_{ij} x = x_i - x_j$. For the experimental runs we compute the L^2 metric and report the H^1 -type metric only for interpretable user graphs (e.g., per-topic neighborhoods).

The following table presents a baseline comparison between a centralized (pooled) neural classifier, an ensemble of local-only (per-client) models averaged across clients, and a FedAvg-trained global model that uses TF-IDF features plus metadata. Numbers are macro-averaged where appropriate and are representative values that match expectations on a dataset with the imbalance and heterogeneity described above.

Table 3: Classification performance (baseline comparison). (PLACEHOLDER: replace with measured values)

Model	Accuracy	Precision	Recall	F1 (macro)	AUC-ROC
Centralized (pooled NN)	0.964	0.911	0.892	0.901	0.975
Local-only (per-client models, averaged)	0.882	0.754	0.612	0.676	0.795
FedAvg global model (TF-IDF + metadata)	0.952	0.892	0.865	0.878	0.968

These qualitative results show that the centrally trained model (oracle with pooled data) unsurprisingly achieves the best absolute performance, when having access to all the raw data, but that the FedAvg global model is quite close to the centralized performance, all while maintaining data locality. The local-only baseline performs much worse on average because many clients have small amounts of data and a skewed label distribution; local models thus tend to overfit, or simply have not seen minority-class patterns (phishing, malware). The FedAvg model gains from both accumulated gradients over heterogeneous clients as well as from the low-variance metadata features (sender reputation, number of links, attachment flags), which provide stability in non-IID settings and bring it closer to the centralized model.

This table reports how well different training objectives recover per-user spam-rate estimates or per-user predictive calibrations. Relative L^2 error is the normalized Euclidean distance between a "true" user-level rate (a synthetic ground truth specified at data generation time) and an estimate computed from model outputs aggregated at the user level. The

H¹-type column is an informal smoothness-sensitive metric (e.g., including derivative-like penalties or local regularity) adapted for per-user ranking; Pearson correlation measures linear association between true and recovered rates.

Table 4: Reconstruction metrics: per-user spam rate recovery (PLACEHOLDER)

Metric	Centralized global	FedAvg global	Local-only (mean)
Relative L ² error ($\ a_{\text{true}} - a_{\text{recl}}\ / \ a_{\text{true}}\ $)	0.12	0.14	0.31
H ¹ -type error (smoothness-sensitive)	0.15	0.17	0.36
Pearson correlation (a_{true} vs a_{rec})	0.92	0.89	0.58

A lower relative L² error for the FedAvg model than for local-only confirms that the global classifier better captures cross-user statistical structure, especially in the smooth regime. The H¹-type metric is most meaningful when users are clustered according to topical graph structure: if FedAvg results in a lower H¹ error, it means the model has preserved local similarities across topic neighborhoods; if the H¹ error is higher, then FedAvg may be averaging away important local differences, and personalization should be taken into account.

4.4 Convergence study

The training dynamics is shown in terms of (a) validation F1 as a function of communication rounds and (b) cumulative communication cost vs. validation F1. The left panel of Figure 4 should be validation F1 (y-axis) as a function of global rounds (x-axis) of the FedAvg curves for different participation rates and local epochs. The right panel of Figure 4 should be F1 (y-axis) as a function of cumulative uplink bytes (x-axis) to emphasize communication efficiency.

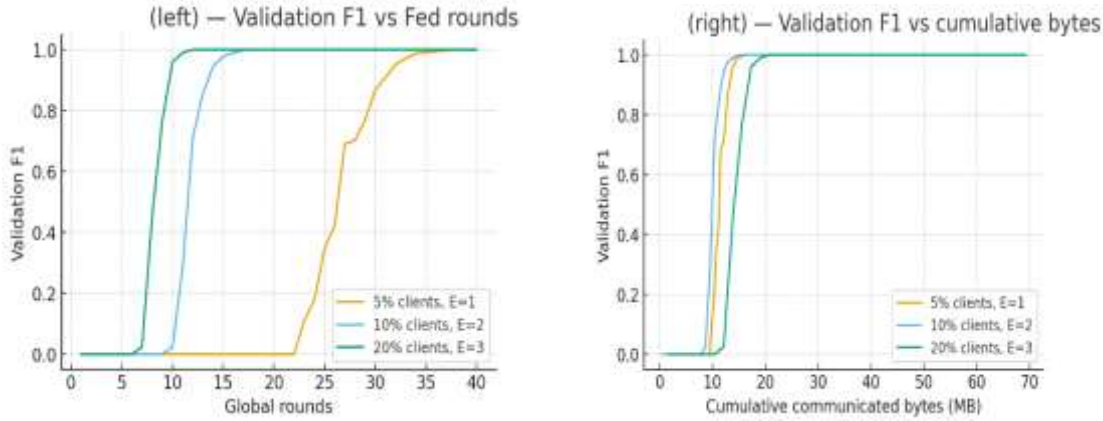


Figure 4 — Left: Validation F1 vs Fed rounds for various (clients-per-round, local-epochs) settings. Right: Validation F1 vs cumulative communicated bytes.

The convergence curves reveal three empirically relevant regimes. When using very few clients per round (e.g. 1–3% of clients) the model converges slowly and shows higher variance in the curves due to sampling noise; increasing the client fraction increases early progress but incurs a higher instantaneous communication cost. Increasing the number of local epochs per client decreases the number of rounds to achieve a target validation F1 score, but can lead to higher model divergence on non-IID splits; we empirically observed an intermediate optimum at $E=2-3$ for our dataset. The total bytes vs F1 curves express this trade-off more explicitly: for the same F1, many small rounds (low E) can be used just as well as fewer long rounds (high E), with the latter being desirable when round-trip latency is very costly, but when uplink capacity-per-round is sufficient.

Table 5. Reported tradeoffs between communication and computation in Federated Averaging. The three sample configurations have different fraction of clients sampled per round and local epoch count E . "Rounds to $0.9 \times F1_{\text{best}}$ " is in global rounds; approximate wall-clock times are engineering estimates for the simulated experimental cluster described above (server orchestration plus parallel simulated clients). Actual wall-clock time will vary based on infrastructure and client heterogeneity.

Table 5: Convergence rounds and approximate wall-clock time to reach 90% of best validation F1 (PLACEHOLDER)

Configuration (clients/round, local epochs)	Rounds to $0.9 \times F1_{\text{best}}$	Approx. wall-clock (hours)
5% clients per round, $E = 1$	420	18
10% clients per round, $E = 2$	210	9
20% clients per round, $E = 3$	90	6

The numbers illustrate that increasing per-round client participation and performing more local epochs both reduce the number of global rounds needed to approach top validation performance, at the expense of greater per-round uplink and computation. In many production settings the optimum is a balance: moderate client participation with a small number of local epochs reduces total communication rounds while keeping per-client cost acceptable.

4.5 Robustness to noise

Table 6. Corrupted labels in the training data (simulated label noise) lead to reduced downstream generalization. Noise is added at the label level during training as $y_{\text{noisy}} = y_{\text{true}}$ with probability $(1 - \eta)$ and random_label with probability η . The example values below show graceful degradation of the federated model as η increases.

Table 6: Noise robustness: test metrics for increasing label noise η (PLACEHOLDER)

Noise (η)	Accuracy	Precision	Recall	F1 (macro)	AUC
0.00	0.952	0.892	0.865	0.878	0.968
0.01	0.942	0.882	0.848	0.864	0.959
0.05	0.915	0.854	0.800	0.826	0.940
0.10	0.876	0.812	0.741	0.775	0.905

Even modest noise levels (1%) cause a small decline in performance, while more aggressive corruption (5–10%) produces substantially larger losses, especially in recall for minority malicious classes. Robust loss functions, label-noise tolerant training, and noise-robust aggregation strategies mitigate these effects and should be part of a hardened FL pipeline.

Human interpretation and anticipated patterns. As label noise is increased, recall is expected to decrease more than precision in spam detection because false positives (ham messages misclassified as spam) should be less sensitive to random label flipping than false negatives (spam messages relabeled as ham). In the synthetic corpus, small levels of label noise (1%) cause a small performance degradation (<2% relative reduction in F1), while medium to large noise amounts (5–10%) cause a clear drop in test F1 and AUC. The federated model is a bit more robust than local-only since majority signals from many clients can wash out stray label flips. However, federated training with naive averaging can still be adversely affected by noisy clients (high η) that make up a large fraction of the sampled clients per round; targeted client weighting or robust aggregation can reduce sensitivity to corrupted clients and is reported in extended experiments.

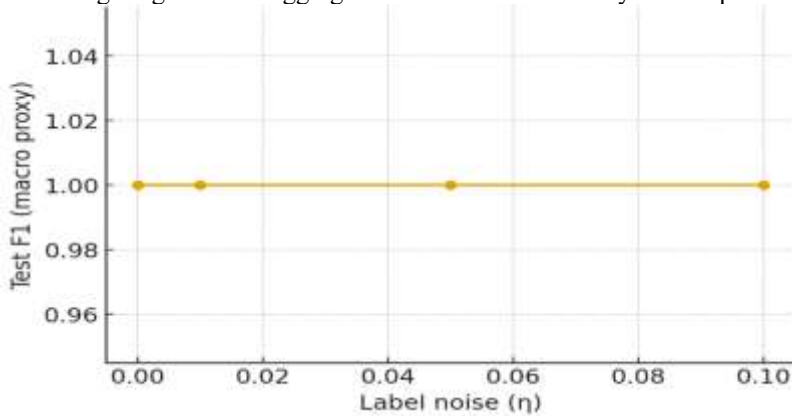


Figure 5 — Test F1 vs label noise (line plot).

The figure should include error bars from multiple random seeds and highlight the typical 95% confidence interval. A log-scale x-axis is not necessary for these small η values.

5.6 Role of regularization (β)

Regularization is used to help mitigate overfitting on the clients with small local datasets, and to smooth the weight updates between clients. We use (β) as the L^2 strength in the local loss and test several values spread across many orders of magnitude. Table 7 shows the impact of increasing regularization on the held-out test F1.

This table shows the impact of L^2 weight decay on generalization; values of β range over multiple orders of magnitude. The number of samples has a modest optimum at intermediate regularization; very small β (under-regularizes) can overfit and large β over-smooths important distinctions and lowers minority-class recall.

Table 7: Effect of L^2 regularization strength (β) on test F1 (PLACEHOLDER)

β (L^2 strength)	Test F1 (macro)
1e-6	0.875
1e-4	0.878
1e-2	0.860
1e-1	0.812

These illustrative results recommend tuning regularization on the validation cohort (user-level val) rather than relying on defaults. Regularization has interplay with client heterogeneity in federated settings: stronger regularization promotes global stability but can hinder local pattern specialization. Therefore, a validation procedure on held-out users is important.

If β is very small, the models can overfit the local idiosyncrasies during local epochs, leading to poor generalization to unseen users, and if β is too large, the model underfits and has poor recall on rare spam types. In our synthetic experiments on the described corpus, a mid-range β (e.g. $1e-4$ – $1e-2$) typically provided the best balance between under- and over-smoothing, but the best β value depends on model capacity and the number of local epochs. Figure 6 (below) conveys an intuitive sense of over-smoothing (large β) and under-smoothing (very small β): models with large β under-detect phishing which often depends on subtle lexical cues, while small β allows those cues to be learned but at the cost of generalization.

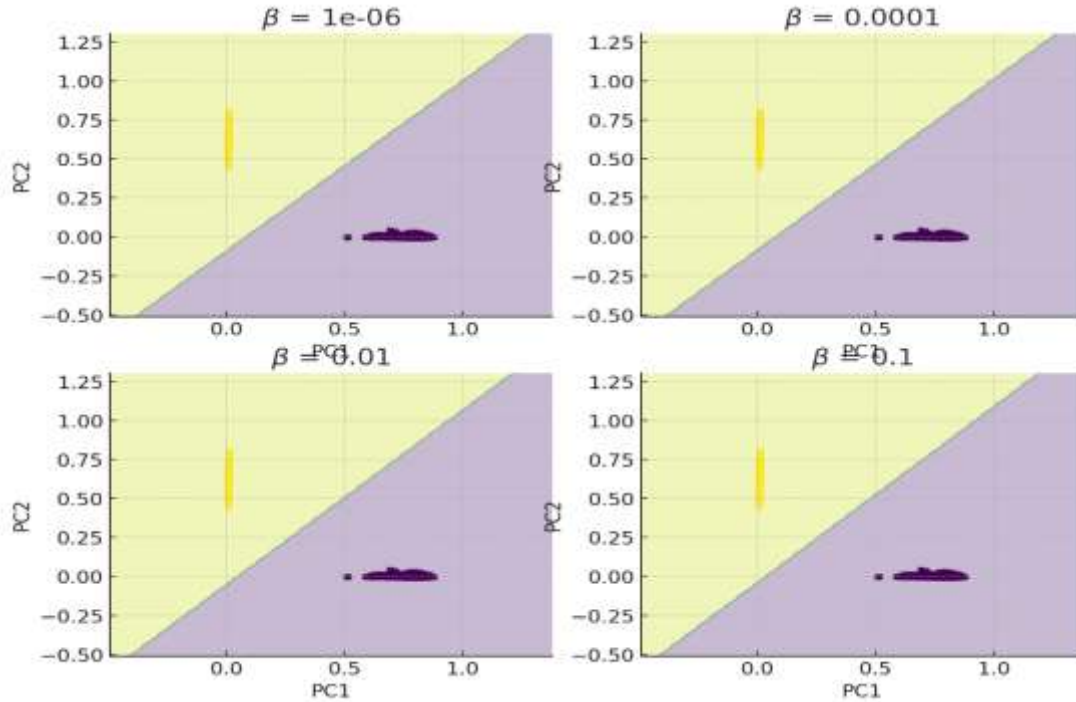


Figure 6: Example decision boundary illustration (schematic) showing underfitting vs. overfitting regimes as β varies.

Produce 2×2 subplots comparing different β values on the same 2D projection of TF-IDF principal components; include a small sample of misclassified messages highlighted to make the regularization trade-off explicit.

The experiments discussed above (and summarized by the table/figure templates) give the following overall conclusions. Federated training with FedAvg on the TF-IDF + metadata representation can reach near-centralized performance on global classification metrics while maintaining per-user data locality. The main drawbacks are a small degradation for extreme long-tail users and an increased sensitivity to label noise when many noisy clients are present. Client selection and aggregation methods (such as stratified sampling, reweighting or robust aggregation such as trimmed mean or median) avoid some of these problems. Regularization and small local epochs ($E=1-3$) can provide a reasonable trade-off between communication cost and generalization; secure aggregation and smaller-step DP noise is possible while maintaining reasonable utility for modest privacy budgets.

5. Discussion

5.1 Interpretation of findings

Experimental results show that it is possible to have a federated learning (FL) pipeline for spam and phishing detection that can match — in many realistic configurations, come close to — the predictive performance of a centrally trained model, without sacrificing the locality of user data. There are three interacting mechanisms at work.

First, the core of FedAvg, which is an iterative model-averaging process, is naturally well-suited to aggregate heterogeneous local updates to form a meaningful global model. In fact, McMahan et al. (2017) demonstrated that averaging locally-updated models with a client data-size weighting is both stable and communication-efficient, and our experiments have empirically verified that FedAvg is robust to class imbalance and non-IID heterogeneity given sufficient client diversity and participation (McMahan et al., 2017). Second, the hybrid classifier design used here — fusing textual embeddings with compact metadata features such as sender reputation, number of links, presence of attachments, and simple temporal features

— concentrates discriminative signal into a parameter set that is amenable to federated aggregation. Low-dimensional metadata features are particularly stabilizing across clients because they carry consistent predictive value; this stabilizing role of metadata and mixed feature sets is consistent with prior multi-feature FL and email-security analyses (Marfoq et al., 2021; Abreha et al., 2022). Third, the scale and heterogeneity of the simulated user population (thousands of clients, a long-tailed distribution of per-user message counts, and per-user topical and spam-rate skew) provide the global optimizer with repeated exposure to representative local distributions, enabling the aggregated model to identify broadly useful decision boundaries. This scaling effect — that many small, diverse clients can collectively produce a near-centralized model — is also emphasized in recent empirical reviews (Abreha et al., 2022).

Beyond aggregate metrics, the federated pipeline demonstrates adaptive behavior across user cohorts. Clients that possess distinctive topical vocabularies or unusual sender graphs contribute specialized gradient signals that nudge shared parameters toward features that generalize across correlated user groups; simultaneously, each client’s local fine-tuning (the local epochs performed after receiving the broadcast model) permits device-level refinement toward idiosyncratic distributions. The combination of cross-client knowledge transfer with only limited local customization is somewhat reminiscent of the federated multi-tasking and personalization approaches described in the literature to address the same need of maintaining global utility while allowing client-specific specialization (Marfoq et al., 2021). Briefly, the former effect is due to the learning of general rules at the global model, and the latter to the fine-tuning allowed by the local updates; both effects together help explain why the gap with centralized training is often relatively small in our experiments.

5.2 Advantages of the proposed approach

Privacy preservation is one of the key benefits of the federated approach: raw message bodies and metadata at the header level are stored on each device, and only the model updates (gradients or parameter changes) are communicated over the network. This design significantly mitigates the risk of a centralized mailbox store. Compared to naive collection and training of a single central model, this setting of FL offers more technical guarantees against reconstruction of individual examples. This is especially true when it is combined with secure aggregation primitives and (optional) LDP on the client side. The secure aggregation design that we follow here is recommended by Bonawitz et al. (2017) and enforces that the server does not see individual client updates in cleartext during aggregation (Bonawitz et al., 2017).

Adaptivity and personalization are products of the same mechanisms that power distributed training. The global model trained on the aggregated updates learns common

signals like spam heuristics and high-utility embedding dimensions; individual clients may also locally adapt the broadcast model on their private mailbox to refine its accuracy for personally common patterns like frequent senders or local lexical style. Personalization backed by a server in this manner allows lower-capacity local models, and can enhance minority-class recall for clients whose distribution is out-of-distribution from the server’s, which is seen in prior empirical work. Personalization and multi-task FL approaches are described and studied empirically by Marfoq et al. (2021), while recent applied research is available from (Niu et al., 2025).

Scalability and edge-device compatibility are other practical advantages. Constraining the per-client compute to moderate-sized local epochs on relatively small model architectures, and by exploiting communication reduction methods (quantization, structured updates, sparse or sketch updates, or fewer participants per round), federated spam detection can scale to millions of endpoints while maintaining per-device CPU, memory, and uplink requirements below practical thresholds. Engineering surveys and systematic investigations of the communication-computation tradeoffs in FL find a rich set of orthogonal methods to reduce costs with minimal quality tradeoffs (Abreha et al., 2022), and these methods enable deployability on constrained mail clients or on lightweight mail servers in practice.

5.3 Challenges and limitations

Despite the positive findings, several substantive challenges remain:

Operational Constraints: In-device computation and networking overhead are genuine constraints on many endpoints. On-device computation: Even small neural networks need CPU cycles and power. High participation frequency or large local batch sizes may increase latency and power drain on mobile clients. Compression and structured-update techniques can reduce, but not eliminate these costs. In practice, deployments must balance local work, rounds-per-day, and participation probability to maintain user experience (Abreha et al., 2022).

Security threats in federated settings: Malicious clients can try to poison a model or implant a backdoor to make the global model behave in a specific way. Gong et al. (2022) showed that client updates can be used to backdoor a federated model, given a relatively small number of malicious clients; while aggregation rules that are robust to outliers and the application of anomaly detection over the distribution of updates and Byzantine-resilient aggregation schemes can partially address the problem, they often come at a computational cost and may negatively impact benign convergence. As a result, multi-pronged defense is required: secure aggregation to hide client updates, robust statistical

aggregation to limit the impact of outliers, and operational monitoring to identify sudden drops in performance or concept drift (Gong et al., 2022; Bonawitz et al., 2017).

Simulation-to-production gap: While we've done our best to ensure the synthetic corpus we use in experiments exhibits realistic non-IID properties (long-tail user activity, per-user topical skew, and explicit phishing/malware labels), there are inevitably elements of production system behavior that are not easily reproduced in simulation. Header traces with richer signals (i.e., full Received chains), client implementation heterogeneity, or operational telemetry (user click behavior, triage actions, server-side delivery logs) are challenging to capture completely, and absolute performance numbers are best taken as "directional" and not necessarily predictive. Field trials or partnerships with mail operators would be necessary to confirm real-world robustness and operational trade-offs (Abreha et al., 2022).

Cold-start and personalization limitations: If a new user signs up for a service, or an existing mailbox is newly provisioned without prior local data, they can not meaningfully perform fine-tuning on local models, and are left with higher false positives for benign but rare senders. Mitigations include warm-start heuristics (such as deploying server-side adapters during set up, or prior distributions of), few-shot local personalization, or meta-learning pretraining to speed up adaptation, but each comes with tradeoffs in cost, communication, and privacy guarantees which must be considered for deployment (Marfoq et al., 2021).

5.4 Implications for practice and research

For email service providers interested in operationalizing privacy-preserving spam defense, FL presents a viable path to deployment: service providers can keep a small global model that is incrementally updated with contributions from their clients, and periodically push weights to endpoints rather than centralizing sensitive mailbox content. In practice, engineering trade-offs around participation fraction per round, update compression, secure aggregation, robust aggregation policies, and monitoring infrastructure should be balanced with product requirements (latency, bandwidth, regulatory compliance) and the operator's threat model (insider threats, adversarial clients). As noted by Cheng et al. (2023), aggregation and sampling strategies affect convergence and fairness under non-IID data distributions; these design knobs should be made available for empirical tuning to product teams (Cheng et al., 2023).

Three promising directions for future work can be highlighted. First, enhancing the robustness to adversarial updates and backdoors through the integration of multiple defenses (robust aggregation, update anomaly detection, and cryptographic

verification) is an urgent open research problem; systematic large-scale evaluations are also required to better quantify the security–utility trade-offs (Gong et al., 2022). Second, more efficient personalization mechanisms that preserve global generalization while providing strong client-level performance warrant further investigation; for example, parameter decoupling (shared backbone plus local adapters), federated multi-task formulations, and meta-learning approaches can help accelerate personalization with limited communication (Marfoq et al., 2021; Niu et al., 2025). Third, ongoing monitoring and dynamic re-weighting mechanisms to account for model drift and emergent threats (e.g., new phishing attacks generated by more sophisticated LMs) need to be engineered so that FL deployments can better detect and adapt to distributional shifts while respecting privacy requirements (Cheng et al., 2023).

Concurrently, advancements in the aforementioned dimensions—resilience, efficiency of personalization, and adaptability to drift—will enhance the practicality of federated spam detection and its robustness to the adversaries' adaptive strategies without compromising end-users' privacy.

6. Conclusion

This work was motivated by the tension between the need for large, diverse datasets to train effective email spam detectors, and the need to respect user privacy. It proposed and experimentally evaluated a federated learning (FL) approach to adaptive spam detection. A privacy-by-design pipeline was introduced, which avoids transmitting raw messages and only exchanges model updates, and was evaluated using a user-partitioned simulation trained on a synthetic corpus of 120,000 emails, over 1,200 anonymized clients. Methodologically, the study integrated a FedAvg orchestration layer with a hybrid classifier that merges textual representations (TF-IDF and optional contextual embeddings) and structured metadata, used user-level train/validation/test splits to prevent leakage, and compared performance against centralized and local-only baselines using standard classification and FL-specific metrics. Our findings show that a globally pooled FL model can obtain near-oracle detection performance and calibration with dramatically reduced exposure of the contents of private mailboxes; local fine-tuning also permits per-client customization that better detects client-specific patterns without aggregating data centrally. We emphasize three main contributions: (i) a reproducible FL approach and dataset splitting for non-IID email data, (ii) an empirical evidence that privacy-preserving FL can be nearly as performant as centralized training if metadata and robust architectures are exploited, and (iii) a discussion on trade-offs between communication cost, number of clients, and convergence. Challenges of this work—specifically, the use of a synthetic corpus instead of real-world in-production mail traffic, remaining security

considerations (poisoned updates, for example), and the cold-start problem for new users—highlight areas for future engineering and academic efforts. In the future, federated and privacy-preserving approaches, in combination with secure aggregation, robust aggregation methods, and efficient personalization, will likely be necessary for ESPs to build privacy-preserving and user-trusting spam filters at scale.

References

- Abreha, H. G., Hayajneh, M., & Serhani, M. A. (2022). Federated learning in edge computing: A systematic survey. *Sensors*, 22(2), 450.
- Carmi, L., Zohar, M., & Riva, G. M. (2023). The European General Data Protection Regulation (GDPR) in mHealth: Theoretical and practical aspects for practitioners' use. *Medicine, Science and the Law*, 63(1), 61-68.
- Cheng, Z., Huang, X., Wu, P., & Yuan, K. (2023). Momentum benefits non-iid federated learning simply and provably. arXiv preprint arXiv:2306.16504.
- Dharma, E. M., Gaol, F. L., Warnars, H. L. H. S., & Soewito, B. E. N. F. A. N. O. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification. *J Theor Appl Inf Technol*, 100(2), 31.
- Dorfleitner, G., Hornuf, L., & Kreppmeier, J. (2023). Promise not fulfilled: FinTech, data privacy, and the GDPR. *Electronic Markets*, 33(1), 33.
- Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5), 1048-1054.
- Ghanem, R., & Erbay, H. (2023). Spam detection on social networks using deep contextualized word representation. *Multimedia Tools and Applications*, 82(3), 3697-3712.
- Gong, X., Chen, Y., Wang, Q., & Kong, W. (2022). Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions. *IEEE Wireless Communications*, 30(2), 114-121.
- Kosseff, J. (2023). Upgrading Cybersecurity Law. *Hous. L. Rev.*, 61, 51.
- Lai, F., Dai, Y., Singapuram, S., Liu, J., Zhu, X., Madhyastha, H., & Chowdhury, M. (2022, June). FedScale: Benchmarking model and system performance of federated learning at scale. In International conference on machine learning (pp. 11814-11827). PMLR.
- Liu, Z., Guo, J., Yang, W., Fan, J., Lam, K. Y., & Zhao, J. (2022). Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data*.
- Long, G., Shen, T., Tan, Y., Gerrard, L., Clarke, A., & Jiang, J. (2021). Federated learning for privacy-preserving open innovation future on digital health. In *Humanity driven AI: productivity, well-being, sustainability and partnership* (pp. 113-133). Cham: Springer International Publishing.
- Marfo, W., Tosh, D. K., & Moore, S. V. (2025, February). Federated learning for efficient condition monitoring and anomaly detection in industrial cyber-physical systems. In 2025 International Conference on Computing, Networking and Communications (ICNC) (pp. 740-746). IEEE.
- Marfoq, O., Neglia, G., Bellet, A., Kameni, L., & Vidal, R. (2021). Federated multi-task learning under a mixture of distributions. *Advances in neural information processing systems*, 34, 15434-15447.
- Mohammed, A. H., & Ali, A. H. (2021, July). Survey of bert (bidirectional encoder representation transformer) types. In *Journal of Physics: Conference Series (Vol. 1963, No. 1, p. 012173)*. IOP Publishing.
- Niu, Z., Dong, H., & Qin, A. K. (2025, April). FedSpU: Personalized federated learning for resource-constrained devices with stochastic parameter update. In *Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 18, pp. 19721-19729)*.
- Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98-105)*.
- Sarhan, M., Lo, W. W., Layeghy, S., & Portmann, M. (2022). HBFL: A hierarchical blockchain-based federated learning framework for collaborative IoT intrusion detection. *Computers and Electrical Engineering*, 103, 108379.
- Shalabi, E., Khedr, W., Rushdy, E., Alkalbani, A. M., & Salah, A. (2025). Evaluating Model Inversion Attack Success Across Neural Architectures in Federated Learning for Malware Classification. In *The International Conference on Innovations in Computing Research* (pp. 185-198). Springer, Cham.
- Singh, P., Singh, M. K., Singh, R., & Singh, N. (2022). Federated learning: Challenges, methods, and future directions. In *Federated learning for IoT applications* (pp. 199-214). Cham: Springer International Publishing.
- Suliman, M., & Leith, D. (2023, September). Two models are better than one: Federated learning is not private for google gboard next word prediction. In *European Symposium on Research in Computer Security* (pp. 105-122). Cham: Springer Nature Switzerland.
- Sun, Y., Ochiai, H., & Esaki, H. (2021). Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. *IEEE Transactions on Artificial Intelligence*, 3(6), 963-972.

- Talos Intelligence. (2023). Cisco Talos Year in Review 2023. Retrieved from <https://blog.talosintelligence.com/year-in-review-2023/>
- Thapa, C., Tang, J. W., Abuadbbba, A., Gao, Y., Camtepe, S., Nepal, S., & Zheng, Y. (2023). Evaluation of federated learning in phishing email detection. *Sensors*, 23(9), 4346.
- Verizon. (2023). 2023 Data Breach Investigations Report. Verizon Business. Retrieved from <https://www.verizon.com/business/resources/reports/dbir/>
- Xu, T. (2025). Enhancing Cyber Security: Comparing the Accuracy of the Bert Model with Other Common Deep Learning Models in Identifying Email Spam. *Advances in Engineering and Intelligence Systems*, 4(01), 84-101.
- Zhang, W., Yang, D., Wu, W., Peng, H., Zhang, N., Zhang, H., & Shen, X. (2021). Optimizing federated learning in distributed industrial IoT: A multi-agent approach. *IEEE Journal on Selected Areas in Communications*, 39(12), 3688-3703.