

Article

SeqFAL: A Federated Active Learning Framework for Private and Efficient Labeling of Security Requirements

Waad Alhoshan 

Department of Computer Science, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia; wmaboud@imamu.edu.sa

Featured Application

SeqFAL enables organizations to collaboratively classify security requirements without sharing sensitive specification documents or proprietary system details. This makes the framework particularly suitable for multi-stakeholder software development environments—such as government suppliers, financial institutions, and regulated industries—where privacy constraints prevent data centralization. *SeqFAL* can be integrated into existing requirements management tools to provide automated, privacy-preserving security analysis while significantly reducing manual labeling effort.

Abstract

Security requirements play a critical role in ensuring the trustworthiness and resilience of software systems; however, their automatic classification remains challenging due to limited labeled data, confidentiality constraints, and the heterogeneous nature of requirements across organizations. Existing approaches typically assume centralized access to training data and rely on costly manual annotation, making them unsuitable for distributed industrial settings. To address these challenges, we propose *SeqFAL*, a communication-efficient and privacy-preserving Federated Active Learning framework for natural language-based security requirements classification. *SeqFAL* integrates frozen pre-trained sentence embeddings, margin-based active learning, and lightweight federated aggregation of linear classifiers, enabling collaborative model training without sharing raw requirement text. We evaluate *SeqFAL* on a combined dataset of SeqReq dataset and the PROMISE-NFR dataset under varying federation sizes, query budgets, and communication rounds, and compare it against three baselines: centralized learning, active learning without federated aggregation, and federated learning without active querying. In addition to the proposed margin-based sampling strategy, we investigate alternative query strategies, including least-confidence and random sampling, as well as multiple linear classifiers such as LinearSVC and SGD-based classifiers with logistic and hinge losses. Results show that *SeqFAL* consistently outperforms FL-only and achieves performance comparable to AL-only centralized baselines, while approaching the optimal upper bound using significantly fewer labeled samples. These findings demonstrate that the joint integration of federated learning and active learning provides an effective and privacy-preserving strategy for security requirements classification in distributed software engineering environments.

Keywords: federated learning; active learning; security requirements; requirements engineering; natural language processing; privacy-preserving machine learning; semantic embeddings; distributed learning; non-IID data; label efficiency



Academic Editor: Douglas O'Shaughnessy

Received: 29 November 2025

Revised: 6 January 2026

Accepted: 7 January 2026

Published: 15 January 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Modern information systems operate in increasingly complex and interconnected environments, making security requirements engineering a crucial discipline for ensuring system robustness, regulatory compliance, and overall trustworthiness [1]. Security requirements articulate the safeguards and constraints necessary to protect software systems against vulnerabilities and cyber threats [2–4]. However, classifying and analyzing such requirements remains a challenging task. Their natural-language form, diverse structure, and domain-specific terminology introduce ambiguity, while significant variability across organizations further complicates automated processing [5,6]. Recent progress in natural language understanding (NLU) has shown promise in supporting this task by improving the interpretation and categorization of textual requirements [7].

Research on automated security requirement classification has evolved through several methodological stages. Initial studies relied on traditional machine learning techniques, followed by deep learning approaches, and more recently transformer-based language models [7]. While these developments have significantly advanced performance, they share two critical shortcomings that limit their applicability in real-world settings. First, these methods depend on large, high-quality labeled datasets, yet annotating security requirements requires domain expertise and is thus slow, costly, and difficult to scale. As a result, most organizations possess limited labeled data—insufficient for training modern NLU models effectively [7]. Second, existing approaches generally assume centralized access to all training data, an assumption that rarely holds in practice. Security requirements often contain confidential architectural details, business processes, internal policies, and other sensitive information that cannot be shared with external parties or even across internal organizational boundaries. This makes centralizing data for model training either impractical or prohibited by security, privacy, and compliance policies [6,8,9].

Federated learning (FL) offers a compelling paradigm to address the second limitation by enabling organizations to collaboratively train models without sharing raw data [10–12]. Instead, each participant trains locally and contributes model updates to a central aggregator. Although FL has demonstrated promising results in domains such as healthcare, finance, and mobile applications [13–15], its application to software and requirements engineering remains limited. Existing FL studies in this domain have often used relatively simple models and have not fully addressed the complexities of natural-language requirements, nor the challenges posed by heterogeneous, non-IID data distributions commonly encountered in organizational settings [16].

Parallel to advances in FL, active learning (AL) provides a complementary strategy to address the first limitation by reducing annotation effort [17,18]. AL prioritizes the most informative unlabeled samples for human annotation, allowing models to improve rapidly with minimal labeled data [19]. However, the majority of AL research assumes a centralized environment where all unlabeled data is pooled together—an assumption incompatible with privacy-sensitive requirements engineering contexts [1]. Integrating AL with FL therefore presents a promising yet underexplored opportunity: enabling organizations to collaboratively identify informative samples while keeping requirement text private and reducing annotation costs [20,21].

To bridge these gaps, this paper introduces *SeqFAL*, a Federated Active Learning framework for Natural Security Requirements. *SeqFAL* combines sentence-level embeddings, lightweight linear classifiers, client-side uncertainty sampling, and federated model aggregation to enable privacy-preserving collaboration across distributed stakeholders. The framework is designed to operate effectively under practical constraints: limited labeled data, heterogeneous requirement distributions, and strict privacy boundaries. *SeqFAL* also includes a robustness mechanism that handles the common scenario where a client

initially possesses labeled examples from only one class, preventing model failures and improving stability during early learning rounds. In the context of this study, a *client* represents an independent organizational unit or development team that maintains its own collection of requirements. Clients do not share raw requirement text with each other or with the central server, reflecting real-world confidentiality and access-control constraints in requirements engineering environments.

To guide this study and structure the evaluation of *SeqFAL*, we formulate the following research questions that reflect the main challenges associated with distributed requirement classification and label-efficient learning:

RQ1: *SeqFAL* Performance Analysis. *How does SeqFAL's performance vary across different linear classifiers, querying strategies, federation sizes, query budgets, and communication rounds?*

The objective of this research question is to systematically analyze the behavior and robustness of *SeqFAL* under a wide range of design and system parameters. Specifically, we investigate (i) the impact of different lightweight linear classifiers, (ii) the effect of alternative querying strategies, (iii) the influence of federation size in increasingly fragmented and heterogeneous client settings, (iv) sensitivity to the query budget per round, and (v) the role of communication rounds in convergence and stability. Through this analysis, we aim to assess the practical reliability of *SeqFAL*.

RQ2: Comparison with Baseline Architectures. *How does SeqFAL compare against established baseline systems, including centralized learning (Upper Bound), federated learning without active querying (FL-NoAL), and active learning without federated aggregation (AL-NoFL), under matched experimental configurations and different linear classifiers?*

The goal of this research question is to quantify the performance gains achieved by integrating federated learning and active learning within a unified framework. By comparing *SeqFAL* to each baseline under equivalent data partitions, label budgets, and classifier choices, we isolate the individual and combined contributions of federation and active querying. This comparison evaluates whether *SeqFAL* consistently outperforms or competes with FL-only and AL-only approaches, while approaching the performance of centralized upper-bound models despite operating under privacy-preserving constraints.

The scope of this work is to study how FL and AL can be effectively combined under realistic non-IID and label-scarce conditions for security requirements identification. The proposed framework emphasizes simplicity and reproducibility in order to isolate the effects of querying strategies, classifier choice, and federated aggregation. Extensions related to adversarial robustness, secure federated learning, and deployment-level system constraints are beyond the scope of this study and are outlined as directions for future work.

To the best of our knowledge, *SeqFAL* represents the first systematic integration of federated learning and active learning for the classification of natural-language security requirements. The main contributions of this work are as follows:

- We propose *SeqFAL*, a novel federated active learning framework tailored for natural-language security requirements, combining semantic embeddings with lightweight and privacy-friendly linear classifiers.
- We introduce a robust client-side uncertainty sampling mechanism with fallback handling for single-class labeled sets, addressing a key challenge in federated requirement engineering environments with non-IID data.
- We conduct a comprehensive evaluation against three baselines: (i) a centralized classifier representing an upper performance bound; (ii) centralized active learning without federated collaboration; and (iii) federated learning without active learning. This evaluation isolates and quantifies the contributions of federation, active learning, and their integration.

- We perform an extensive ablation study varying the linear classifiers, query sampling strategies, number of clients, query sizes, and communication rounds, complemented by non-parametric significance tests. Results demonstrate that *SeqFAL* consistently competes both AL-only and FL-only baselines and approaches centralized performance despite operating under strict privacy constraints.

The remainder of the paper is organized as follows. Section 2 reviews relevant work in requirements engineering, federated learning, and active learning. Section 3 presents the *SeqFAL* framework. Section 4 details the experimental setup. Section 5 reports and analyzes the findings. Section 6 discusses the key findings and practical implications followed by threats to validity. Section 7 concludes the study and identifies avenues for future research.

2. Background and Related Work

2.1. Security Requirements Classification

Requirements classification is a foundational task in requirements engineering (RE), enabling the organization, prioritization, and automated understanding of natural-language software specifications. Early approaches were largely rule-based or keyword-driven, relying on handcrafted patterns and domain heuristics. Although effective for constrained datasets, these methods lacked generalizability across organizational contexts and were brittle when confronted with linguistic variability [1]. To overcome these limitations, supervised machine learning techniques were introduced, demonstrating improved robustness and generalization [7]. More recent advances in deep learning and transformer-based language models have further enhanced performance on multi-domain RE datasets [22].

Security requirements have emerged as a distinct and vital focus within requirements engineering, driven by the increased prevalence of cyber threats and the complexity of today's software systems [3]. In earlier decades, security was often implicitly included within general non-functional requirements or quality attributes. However, as the risk associated with inadequate security became apparent through incidents and failures, the software engineering community recognized the need for explicit identification and separate classification of security requirements [4].

This shift led to the development of dedicated frameworks, standards, and methodologies focused on systematically distinguishing security-related requirements from other types during requirements engineering [7]. Prior work has explored semantic similarity approaches [2], supervised classification [5,6], zero-shot learning [23], and explainable neural architectures [24]. These studies highlight that distinguishing security requirements from other functional or non-functional statements requires complex semantic representations. However, most methods assume access to a centrally collected and labeled dataset—an assumption that is difficult to satisfy in practice due to confidentiality constraints. Requirements often contain sensitive architectural details and policy information that cannot be shared across teams or organizations. As a result, centralized training pipelines may conflict with security and privacy policies that govern how requirements are stored and accessed.

From a practical perspective, security requirements classification is most commonly formulated as a binary decision problem, distinguishing security-related requirements from non-security ones [1,7]. In industrial RE practice, security is typically treated as a dedicated category within non-functional requirements [25], and early identification often focuses on determining whether a requirement raises security concerns at all, rather than assigning it to one of many fine-grained categories. While multi-class formulations have been explored in academic settings—often spanning multiple non-functional requirement types—such settings introduce additional annotation complexity, class imbalance, and ambiguity, which can limit applicability in real-world industrial contexts [2]. Consequently, binary classifi-

cation remains the predominant formulation in practice, particularly in scenarios where labeling resources are limited and requirements data are sensitive.

2.2. Active Learning for Requirements Classification

Active learning (AL) aims to reduce labeling effort by selectively querying the most informative unlabeled requirements for human annotation, and classical strategies such as uncertainty sampling, query-by-committee, and expected model change have demonstrated substantial gains in label efficiency across NLP and requirements-related text classification and sequence-labeling tasks [18,19,21]. In requirements engineering specifically, approaches like MARE show that AL can significantly reduce the amount of labeled data needed to train effective classifiers for functional and non-functional requirements [20], while recent work on AL with pre-trained language models for NER in RE further highlights the benefits of combining modern NLP with query strategies tailored to expert annotation workflows [21]. AL is therefore particularly attractive in domains that demand expert annotators—such as security requirements engineering—where annotation is costly and specialized, making selective querying a natural fit for requirements engineering applications [20,21].

However, most existing AL research implicitly assumes access to a centralized pool of unlabeled data that can be freely queried by a single learner, an assumption that conflicts with real-world requirements engineering practice, where requirements are fragmented across teams, tools, projects, and organizational boundaries, and where confidentiality and regulatory constraints often preclude aggregating raw artifacts into a single repository [18,21]. In practice, organizations increasingly adopt distributed and federated development settings, which further restrict direct sharing of requirements and limit the applicability of conventional pool-based AL setups, thereby motivating privacy-preserving and communication-aware extensions of AL that can operate without centralizing data.

Moreover, many classical AL techniques presuppose that the initial labeled set contains examples from all target classes. In security requirements classification, non-security requirements typically dominate, so local datasets at the team or client level can be highly imbalanced or effectively one-class, undermining the reliability of uncertainty estimates and the effectiveness of standard query strategies [20,21]. These conditions motivate the development of more robust, class-imbalance-aware, and privacy-preserving active learning mechanisms that operate in federated settings and remain effective under skewed label distributions and limited local supervision [18].

2.3. Federated Learning for Text Classification

Federated learning (FL) provides a decentralized alternative to centralized model training by enabling clients to collaboratively build models without sharing raw data. This paradigm has been widely adopted in privacy-critical domains such as healthcare, finance, and mobile personalization [13,14,26]. FL has also gained traction in natural language processing (NLP), supporting tasks such as text classification and next-word prediction while preserving data locality [27]. Despite this progress, research on FL for requirements engineering is still in its infancy. A recent study proposed FL for distributed requirements classification [16], demonstrating the feasibility of training models across fragmented requirement repositories. Related federated NLP work commonly incorporates neural architectures or model distillation techniques [28,29], but these approaches often involve substantial computational overhead and communication costs.

Furthermore, federated NLP systems frequently assume balanced or identically distributed client data. In practice, requirement datasets across organizational units are inherently non-IID; teams write requirements in different styles, domains, and levels of detail. This linguistic heterogeneity poses challenges for neural FL models, which may

converge slowly or exhibit degraded performance. Lightweight linear models and semantic embeddings offer a promising alternative in such settings, as they require fewer resources and may generalize better under non-IID distributions. However, their potential for security requirement classification in federated environments remains largely unexplored.

2.4. Research Gap

The existing works reveals a persistent disconnect between real-world constraints in RE and the assumptions embedded in current security requirement classification methods. Most prior work relies on centralized datasets, overlooking the fact that natural-language requirements frequently contain sensitive or proprietary information that cannot be pooled due to confidentiality constraints (e.g., [24,30]). At the same time, effective annotation of security requirements demands expert knowledge, creating a scarcity of labeled data and elevating annotation costs—conditions under which fully supervised models, including recent deep or transformer-based approaches, struggle to operate [1,7]. Although FL offers a pathway to preserving data privacy and AL can reduce labeling effort, their combined potential has rarely been explored in the context of RE. Existing FL studies tend to focus on computationally heavy neural models, while AL research generally presumes centralized access to data, leaving open questions about how these techniques perform under realistic challenges such as non-IID client distributions or clients with only single-class labeled samples (e.g., [16]). These gaps together underscore the need for a lightweight, label-efficient, and privacy-preserving approach tailored to distributed natural-language security requirements, motivating the development of *SeqFAL*, the framework proposed in this study.

3. Proposed Framework (*SeqFAL*)

This section introduces *SeqFAL*, a Federated Active Learning framework designed to classify natural-language security requirements in distributed, privacy-sensitive environments. *SeqFAL* integrates lightweight semantic representations, margin-based active learning, and parameter-based federated aggregation to support collaborative model development without sharing raw requirement text or relying on large annotated corpora. The design of *SeqFAL* reflects realistic industrial constraints, including fragmented data sources, confidentiality restrictions, heterogeneous distributions of requirement types, and limited availability of labeled examples. The overall workflow of the proposed framework is depicted in Figure 1 and in the following we explain the detailed steps.

Each client maintains its own repository of requirements and transforms them into dense semantic vectors using a shared, pre-trained sentence encoder. For a requirement sentence s , its embedding is computed locally through a frozen encoder:

$$\mathbf{x} = f_{\theta}(s), \quad (1)$$

where f_{θ} denotes the encoder with fixed parameters θ . This representation captures contextual semantics while ensuring privacy, since no raw text or gradients are transmitted externally.

Using these embeddings, each client trains a lightweight **linear classifier** to distinguish between security and non-security requirements. This design supports efficient local computation and stable binary decision boundaries in high-dimensional embedding spaces. For labeled data $(\mathbf{X}_i, \mathbf{y}_i)$ on client i , the classifier parameters $(\mathbf{w}_i, b_i)$ are learned by minimizing a margin-based objective:

$$\min_{\mathbf{w}_i, b_i} \mathcal{L}(\mathbf{w}_i, b_i) + \lambda \|\mathbf{w}_i\|^2, \quad (2)$$

where $\mathcal{L}(\cdot)$ is a binary loss function encouraging correct separation and λ controls regularization (The specific classifier instantiation is presented later in the Experimental Setup).

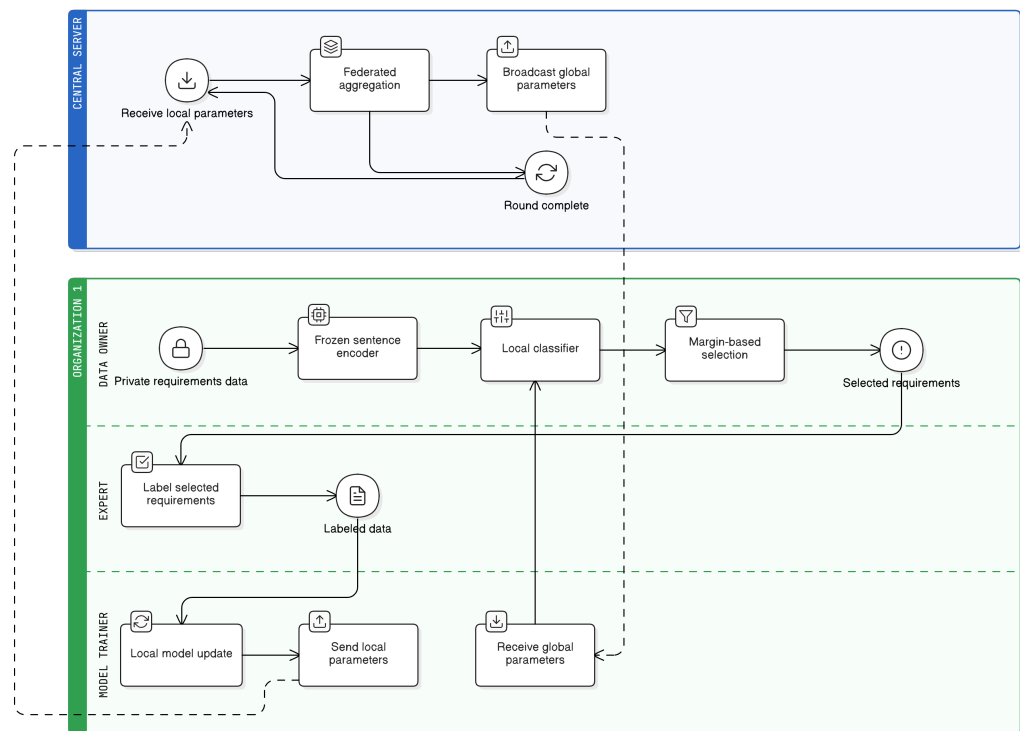


Figure 1. Overview of the *SeqFAL* workflow, illustrating a single organization as a representative federated client. Each organization performs local active learning by selecting uncertain requirements, obtaining expert labels, and updating a local classifier using frozen sentence embeddings, while all raw data remain on-premise. Only local model parameters are shared with the central coordinator for aggregation and broadcast, enabling privacy-preserving federated learning across communication rounds.

A central component of *SeqFAL* is the integration of **active learning** directly on each client to reduce annotation cost. After training the local classifier, each client computes an uncertainty score for every unlabeled requirement. *SeqFAL* uses margin-based uncertainty sampling, where the informativeness of an unlabeled instance $\mathbf{x}$ is defined as the absolute distance between its decision value and the classifier boundary:

$$u(\mathbf{x}) = 1 - \left| \mathbf{w}_i^\top \mathbf{x} + b_i \right|. \quad (3)$$

A larger value of $u(\mathbf{x})$ indicates greater uncertainty (i.e., $\mathbf{x}$ lies closer to the decision boundary). Each client selects the top- q unlabeled instances with the highest uncertainty score:

$$\mathcal{Q}_i = \arg \max_{\mathbf{x} \in \mathcal{U}_i} u(\mathbf{x}). \quad (4)$$

This identifies the requirements most likely to improve the classifier when labeled. In realistic RE settings, some clients may initially possess labeled data from only one class; in such cases, uncertainty scores are not meaningful. *SeqFAL* addresses this by falling back to random sampling until both classes appear, ensuring that the classifier remains well-defined. After the labeled set contains more than one class, margin-based uncertainty sampling is resumed in subsequent rounds.

Once selected, the queried requirements are labeled by the local expert, added to the client's labeled set, and removed from the unlabeled pool. The local classifier is retrained, and only the updated model parameters $(\mathbf{w}_i, b_i)$ are transmitted to the central server. At no stage are requirement texts or embeddings shared.

SeqFAL employs a communication-efficient form of parameter aggregation inspired by federated learning. Specifically, it adopts a weighted parameter averaging strategy (FedAvg) that is widely used in federated learning for linear and shallow models under convex objectives [12,31]. The global model is constructed as a weighted average of client parameters:

$$\mathbf{w}_{\text{global}} = \sum_{i=1}^K \frac{n_i}{N} \mathbf{w}_i, \quad b_{\text{global}} = \sum_{i=1}^K \frac{n_i}{N} b_i, \quad (5)$$

where K is the number of clients, n_i is the number of labeled examples on client i , and $N = \sum_{i=1}^K n_i$. Although this aggregation is not theoretically equivalent to optimizing a centralized SVM objective, *SeqFAL* intentionally operates in a restricted and controlled setting with frozen sentence embeddings, linear classifiers, and identical hyperparameters across all clients. Under these conditions, the learning problem is convex and weighted parameter averaging serves as a practical and empirically stable approximation that is commonly adopted in federated learning practice. This aggregated model is redistributed to all clients, replacing their local parameters and serving as the starting point for the next active learning round.

Algorithm 1 summarizes how *SeqFAL* integrates federated learning and active learning in a unified training loop. In *SeqFAL*, each communication round integrates both active learning and federated aggregation. Specifically, during a single round, each client (1) embeds its local requirements, (2) checks whether its labeled data contain more than one class and, if so, trains a lightweight linear classifier, (3) applies margin-based active learning to identify the most informative unlabeled requirements when class diversity is available (otherwise resorting to random sampling), (4) queries labels and updates its local dataset, and (5) transmits the updated model parameters to the central server; if a client remains single-class after sampling, it temporarily reuses the current global model parameters. The server then aggregates the received parameters to form a new global model, which is broadcast back to all clients for the next round. This iterative process enables collaborative model improvement across distributed clients while preserving data privacy and reducing manual labeling effort.

Algorithm 1 *SeqFAL*: Federated Active Learning for Security Requirements Classification

Require: Clients $\{1, \dots, K\}$, encoder f_θ , query size q , rounds T

Ensure: Global classifier $(\mathbf{w}_{\text{global}}, b_{\text{global}})$

- 1: Initialize $(\mathbf{w}_{\text{global}}, b_{\text{global}})$ randomly
 - 2: **for** $t = 1$ to T **do**
 - 3: **for** each client i **in parallel do**
 - 4: Embed requirements: $\mathbf{x} = f_\theta(s)$ for all $s \in \mathcal{D}_i$
 - 5: **if** both classes appear in $\mathcal{L}_i$ **then**
 - 6: Train local linear classifier on $\mathcal{L}_i$ to obtain $(\mathbf{w}_i, b_i)$
 - 7: Compute margins $m(\mathbf{x}) = |\mathbf{w}_i^\top \mathbf{x} + b_i|$
 - 8: Select q smallest-margin samples
 - 9: **else**
 - 10: Randomly sample q items from $\mathcal{U}_i$ to bootstrap class diversity
 - 11: **end if**
 - 12: Query labels and update $\mathcal{L}_i, \mathcal{U}_i$
 - 13: Retrain classifier on updated $\mathcal{L}_i$; send $(\mathbf{w}_i, b_i, n_i)$ to server
 - 14: **end for**
-

Algorithm 1 *Cont.*

15: Aggregate global model:

$$\mathbf{w}_{\text{global}} = \sum_i \frac{n_i}{N} \mathbf{w}_i, \quad b_{\text{global}} = \sum_i \frac{n_i}{N} b_i$$

16: Broadcast $(\mathbf{w}_{\text{global}}, b_{\text{global}})$ to clients17: **end for**18: **return** $(\mathbf{w}_{\text{global}}, b_{\text{global}})$

Next, we detail the experimental setup used to assess the effectiveness of *SeqFAL* compared with established baselines.

4. Experimental Design

This section describes the experimental setup used to evaluate *SeqFAL* and compare it against multiple baseline systems. The design covers dataset preparation, client partitioning, training configuration, evaluation metrics, baselines, and implementation details. All settings correspond to the implementation used in our publicly shared code repository.

Datasets. We use the *SeqReq* dataset which was originally Created by Knauss et al. [2,32], this dataset contains 510 requirements, made of security-related requirements (187) and non-security related requirements (323). The requirements were collected from three industrial projects: Common Electronic Purse (ePurse), Customer Premises Network (CPN), and Global Platform Spec (GPS). The dataset has been used, e.g., by Varenov and Gabdrahmanov [5] and by Kobilica et al. [33]. In addition, we extended the experimental datasets by incorporating requirements from the PROMISE dataset, originally developed by Cleland-Huang et al. [25,34]. The PROMISE dataset comprises more than 13 software projects and collectively contains 369 non-functional requirements (NFRs). Among these, 66 requirements are labeled as security-related, while the remaining instances are distributed across other non-functional categories, such as usability and performance. For our experiments, the dataset was converted into a binary classification setting, where security-related requirements constitute the positive class and all remaining NFRs are grouped into a single non-security class. The PROMISE dataset has been widely used in prior work, for example by Zadenoori et al. [35].

Non-IID partitioning. To simulate realistic federated scenarios in which requirements are distributed across organizational units, we partition the dataset into K non-overlapping client datasets using a reproducible *Dirichlet label-skew* mechanism. For each class, client mixing proportions are sampled from a Dirichlet distribution with concentration parameter $\alpha = 0.3$ using a fixed random seed (42), and class instances are allocated accordingly. This procedure induces heterogeneous, non-IID class distributions across clients, with varying proportions of security and non-security requirements. We vary the number of clients $K \in \{2, 3, 4, 5, 6, 7, 8, 9\}$ to simulate different degrees of data decentralization and fragmentation. Each client receives a distinct subset of requirements, which is split into an initial labeled seed of two examples and a remaining unlabeled pool for active querying. In each communication round, clients query $q \in \{1, 2, 3, 4, 5\}$ unlabeled instances and add the newly annotated examples to their local labeled sets. Due to the non-IID partitioning, the class composition of the initial labeled set may vary across clients, and some clients may initially contain examples from only a single class. The labeled set then grows over successive rounds as clients acquire additional labels through active querying.

Training procedure. The training procedure of *SeqFAL* consists of three main stages that are executed iteratively within each communication round: requirement encoding, local linear classification, and federated active learning with model aggregation. Table 1

summarizes the principal parameters used in all experiments. Prior to training, the dataset is split into a fixed training pool (70%) and a held-out test set (30%). All active learning queries, local model updates, and federated aggregations are performed exclusively on the training pool, while the test set remains completely untouched—never queried, labeled, or used for training—across all active learning rounds and communication cycles, and is used only for final evaluation.

Requirements Encoding. All requirement sentences are converted into dense semantic vectors using a pre-trained MiniLM encoder, consistent with the embedding stage described earlier. The selection of this particular encoder was motivated by prior empirical evidence: a comparative study evaluating both general-purpose and domain-specific sentence encoders in zero-shot settings for requirements classification showed that MiniLM consistently achieved top performance in both binary and multi-class classification tasks [23]. To be more specific, prior evidence (e.g., [23,36]) suggests that general-domain encoders are often better suited for requirements classification due to the diverse terminology across systems and domains, while domain-specific models may offer limited coverage of cross-domain terminology.

To ensure a fair and controlled evaluation of the learning dynamics, the encoder component of *SeqFAL* remains frozen throughout all experiments, and embeddings are computed once and reused across all configurations.

Linear Classification. Each client trains a lightweight linear classifier on its local labeled data. In this step of *SeqFAL*, we experiment with three linear classification models that are well-suited for federated and low-resource settings: Linear Support Vector Classification (LinearSVC), an SGD-based classifier with hinge loss—linear SVM—(SGD-Hinge), and an SGD-based classifier with log loss—logistic regression—(LogReg). These models are selected due to their computational efficiency, convex optimization properties, and compatibility with federated averaging. Model parameters are re-trained locally whenever new labels are acquired through active querying.

Sampling Strategy in Active Learning. *SeqFAL* primarily employs margin-based uncertainty sampling, as described earlier, which selects unlabeled instances closest to the decision boundary and is well-suited for linear classifiers. To contextualize the effectiveness of margin-based querying, we additionally consider two alternative sampling strategies: random sampling, which serves as a lower-bound baseline without uncertainty awareness, and least-confidence sampling, which selects instances for which the classifier assigns the lowest maximum class probability. It is important to note that least-confidence sampling relies on calibrated class probability estimates, which are not natively provided by some linear classifiers used in this study (e.g., LinearSVC). Accordingly, least-confidence sampling is applied only when reliable probability estimates can be obtained through probabilistic calibration, ensuring that uncertainty scores remain meaningful and comparable across configurations. These strategies are included to provide a controlled comparison against the proposed margin-based approach and to assess whether the choice of uncertainty criterion materially influences performance under federated and label-scarce conditions.

Federated and Active Learning Parameters. *SeqFAL* is evaluated across multiple active learning budgets with query sizes $q \in \{1, 2, 3, 4, 5\}$ per communication round, and across communication rounds $T \in \{5, 10, 15, 20\}$. At each round, every client applies sampling technique to select unlabeled instances for annotation when sufficient class diversity is available; otherwise, random sampling is used as a fallback to avoid single-class training. For each configuration, the full *SeqFAL* loop is executed, including local training on the current labeled set, the sample selection, simulated annotation, model re-training, and federated aggregation of client models via weighted averaging.

Table 1. Training and evaluation configuration used in all experiments.

Parameter	Value/Range
Sentence encoder	MiniLM (general-domain)
Classifier type C	LinearSVC, SGD-based linear SVM, SGD-based logistic regression
Query size q	{1, 2, 3, 4, 5}
Communication rounds T	{5, 10, 15, 20}
Number of clients K	{2, 3, 4, 5, 6, 7, 8, 9}
Sampling strategy S	Margin, Random, Least-confidence
Federated aggregation	Weighted parameter averaging (FedAvg rule)
Train/test split	Strict held-out test set (70/30 split; never queried or used in training)
Random seeds	10 runs (initial seed = 42, incremented by 1)

Ablation protocol note. To summarize the aforementioned training procedure, a strictly controlled experimental protocol is adopted in which federated learning and active learning are always applied jointly. This means that all sampling strategies and federated configurations are evaluated within the same unified FL and AL setting, rather than in isolation. The ablation study systematically examines different design choices, including federation size ($K = 8$), number of communication rounds ($T = 4$), sampling strategy ($S = 3$), query batch size ($q = 5$), and classifier type ($C = 3$). Under this protocol, a total of 1440 experimental configurations are conducted for each random seed.

Evaluation metrics. To assess model performance, we report accuracy, weighted precision, weighted recall, and weighted F1-score. Accuracy measures the proportion of correctly classified requirements and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively.

Because the SeqReq dataset is class-imbalanced, we additionally compute weighted metrics, where each class contributes proportionally to its support. For classes $c \in \{1, 2\}$ with n_c samples and total size $N = \sum_c n_c$, weighted precision, recall, and F1-score are defined as:

$$\text{wP} = \sum_c \frac{n_c}{N} \cdot \frac{TP_c}{TP_c + FP_c} \quad (7)$$

$$\text{wR} = \sum_c \frac{n_c}{N} \cdot \frac{TP_c}{TP_c + FN_c} \quad (8)$$

$$\text{wF1} = \sum_c \frac{n_c}{N} \cdot \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (9)$$

Weighted metrics ensure that the contribution of each class reflects its frequency, providing a robust evaluation of *SeqFAL* under class imbalance. All metrics are computed on the global model obtained after the final communication round.

Statistical significance tests. To evaluate whether key factors within *SeqFAL*—namely the choice of linear classifier (C), the query sampling strategy (S), the number of clients (K), query size (q), and communication rounds (T)—have a significant impact on performance, we conduct non-parametric statistical tests across all experimental configurations. Because the metric distributions do not satisfy normality assumptions and the experiments involve repeated measures across multiple *SeqFAL* settings, we apply the Friedman test to assess overall differences among factor levels for each evaluation metric. When the Friedman test indicates significance, we perform post-hoc pairwise comparisons using the Nemenyi test to identify which specific configurations differ reliably. This analysis provides a principled assessment of how FL- and AL-related parameters influence *Seq-*

FAL's effectiveness and helps quantify which design choices most strongly contribute to performance improvements in the proposed framework. Each statistical analysis treats a single experimental configuration—defined by a fixed combination of all factors except the one under study—as one statistical block. For each block, performance is first aggregated by averaging over independent random seeds, ensuring that repeated runs of the same configuration are not treated as separate samples. A non-parametric Friedman test is then applied across these seed-averaged blocks to assess whether systematic performance differences arise from varying the factor of interest. When the Friedman test indicates statistical significance, Kendall's W is additionally reported to quantify the strength of the observed effect, providing an interpretable measure of effect size across blocks. This design avoids pseudo-replication by preventing correlated measurements (e.g., multiple seeds) from being counted as independent observations. *The outcomes of these analyses support answering RQ1.*

Baselines. *SeqFAL* is evaluated against three baseline conditions to contextualize its performance:

- *Centralized Upper-Bound:* All requirements are combined into a single dataset and used to train a linear binary classifier with full labeled supervision. This represents an optimistic upper-bound scenario with no privacy or distributional constraints.
- *Active Learning without Federated Learning (AL-NoFL):* Active learning is applied centrally using the same embedding model and margin-based uncertainty sampling strategy originally proposed in *SeqFAL*, but without any federated partitioning. This isolates the contribution of federated aggregation by controlling for AL effects.
- *Federated Learning without Active Learning (FL-NoAL):* Clients train the same linear classifier under a standard federated setup using the same non-IID data partitioning controlled by the Dirichlet parameter α as in *SeqFAL*, but rely solely on their initial labeled samples with no active querying. This isolates the contribution of AL by controlling for the federated component.

All baselines use identical embedding representations and classifier configurations to ensure a fair comparison. These baselines enable us to disentangle the individual contributions of federated learning, active learning, and their combination within *SeqFAL*. For consistency, we use the term 'round' (T) to refer to each query–retrain cycle in AL-NoFL and each communication step in FL-based settings, enabling comparable configuration settings across all baselines. *The results of comparing the best-performing configuration of SeqFAL with the aforementioned baselines guide the analysis in RQ2.*

Multiple seeds experimental protocol. All experiments were repeated over multiple random seeds to account for stochasticity arising from data partitioning, active learning sampling, and model optimization. Specifically, we conducted each experimental setting over ten independent runs using random seeds initiating from 42. Each seed fully determines the client data splits, query selections, and training randomness. For a given seed, all compared methods were evaluated under identical conditions, enabling paired comparisons across methods. Performance is aggregated across seeds and summarized using descriptive statistics, including the mean and standard deviation, as well as 95% confidence intervals.

Implementation environment. All experiments were conducted using Python (version 3.12) in Google Colab, leveraging scikit-learn (version 1.5), NumPy (Version 2.0), and sentence-transformers. Ablation studies and significance tests were executed entirely in the provided Colab notebooks. To support transparency and reproducibility, the full implementation of *SeqFAL*—including dataset preparation, federated simulation scripts, baseline models, active learning routines, and evaluation code—is publicly available at: <https://github.com/waadalhoshan/SeqFAL> (accessed on 29 November 2025). This

repository contains all scripts and notebooks required to reproduce the results presented in this work.

5. Results Analysis

This section presents the empirical findings of our study, organized around the research questions defined earlier.

5.1. Answers to RQ1: SeqFAL Performance Analysis

To answer RQ1, we analyze SeqFAL's performance across five experimental factors: classifier choice (C), querying strategy (S), federation size (K), query size (q), and communication rounds (T). For each factor, performance is summarized using mean $\pm$ std, as well as 95% confidence intervals, over repeated random seeds, and trends are illustrated through the corresponding figures and summary tables.

Effect of Linear Classifier. Figure 2a shows the evolution of **accuracy** across communication rounds for the three linear classifiers: Linear Support Vector Classification (LinearSVC), an SGD-based classifier with hinge loss (SGD-Hinge), and an SGD-based classifier with log loss (LogReg). All classifiers exhibit consistent improvements as the number of communication rounds increases, indicating stable convergence behavior within SeqFAL. Across all rounds, logistic regression (LogReg) maintains higher accuracy values, while LinearSVC and SGD-Hinge follow similar trajectories with slightly lower performance.

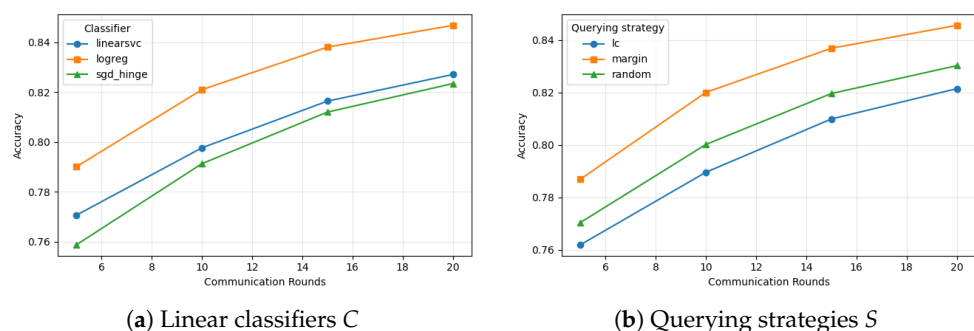


Figure 2. Accuracy trends in SeqFAL across communication rounds. Results are averaged over federation sizes, query budgets, and random seeds.

Table 2 reports classifier-level performance in terms of weighted F1-score ($wF1$), weighted precision (wP), and weighted recall (wR). Results are first averaged within each random seed and then aggregated over 10 independent runs; we report mean $\pm$ standard deviation together with 95% confidence intervals computed over seed-level averages. Logistic regression achieves the highest overall performance across all three metrics, with $wF1 = 0.834 \pm 0.027$ (95% CI [0.815, 0.854]), $wP = 0.855 \pm 0.018$, and $wR = 0.847 \pm 0.020$. LinearSVC and SGD-Hinge yield slightly lower but closely aligned results, indicating comparable precision–recall trade-offs across these classifiers.

Table 2. Classifier-level performance in SeqFAL. Results are averaged within each random seed and then aggregated over 10 independent runs. We report mean $\pm$ standard deviation and 95% confidence intervals for weighted F1-score ($wF1$), weighted precision (wP), and weighted recall (wR).

Classifier	$wF1$ (Mean $\pm$ Std)	$wF1$ (95% CI)	wP (Mean $\pm$ Std)	wR (Mean $\pm$ Std)
LogReg	0.834 ± 0.027	[0.815, 0.854]	0.855 ± 0.018	0.847 ± 0.020
LinearSVC	0.803 ± 0.038	[0.776, 0.830]	0.837 ± 0.025	0.827 ± 0.027
SGD-Hinge	0.800 ± 0.030	[0.778, 0.821]	0.836 ± 0.022	0.823 ± 0.020

To assess whether classifier-dependent differences are consistent across experimental contexts, we applied a Friedman test on $wF1$, treating the classifier as the factor of interest, while using K , q , rounds, and query as blocking variables. The test indicates a statistically significant effect of the classifier on $wF1$ ($\chi^2 = 572.85$, $p < 0.001$). The associated effect size, measured using Kendall's W , is $W = 0.597$, which corresponds to a *large* effect, indicating strong and consistent differences in classifier rankings across blocks.

Post-hoc Nemenyi pairwise comparisons further show that logistic regression differs significantly from both LinearSVC and SGD-Hinge across blocks, while the difference between LinearSVC and SGD-Hinge is also statistically significant. Taken together, these results suggest that classifier choice has a *systematic and consistent impact* on SeqFAL's performance. While absolute differences in central tendency remain moderate, the large effect size indicates that the relative ranking of classifiers is stable across a wide range of experimental configurations.

Effect of Querying Strategy. Figure 2b illustrates the evolution of **accuracy** across communication rounds for the different querying strategies. All strategies exhibit a consistent upward trend as training progresses, indicating stable learning behavior within SeqFAL. Margin sampling consistently achieves higher accuracy values across all communication rounds, while random sampling and least-confidence sampling follow similar trajectories with slightly lower performance. The separation between curves remains relatively stable, suggesting that querying strategy influences performance trends but does not lead to divergent learning dynamics.

Table 3 reports querying-strategy-level performance in terms of weighted F1-score ($wF1$), weighted precision (wP), and weighted recall (wR). Results are first averaged within each random seed and then aggregated over 10 independent runs; we report mean $\pm$ standard deviation together with 95% confidence intervals computed over seed-level averages. Margin sampling (i.e., FL-MarginAL), which constitutes the original querying strategy proposed in SeqFAL, achieves the highest overall performance with $wF1 = 0.831 \pm 0.031$ (95% CI [0.808, 0.853]), followed by FL-RandomAL (0.811 ± 0.030 , 95% CI [0.789, 0.832]) and FL-LeastConfidenceAL (0.795 ± 0.034 , 95% CI [0.771, 0.820]). The higher F1-score of margin sampling (i.e., FL-MarginAL) is accompanied by consistent gains in both weighted precision and weighted recall.

To assess whether these differences are consistent across experimental contexts, we applied a Friedman test on $wF1$, treating the querying strategy as the factor of interest and using K , q , rounds, and classifier as blocking variables. The test indicates a statistically significant effect of the querying strategy ($\chi^2 = 430.38$, $p < 0.001$). The associated effect size, measured using Kendall's W , is $W = 0.448$, corresponding to a *moderate* effect, indicating that differences in querying strategies are systematic but less pronounced than those observed for the classifier factor. Post-hoc Nemenyi pairwise comparisons further show that margin sampling differs significantly from both random and least-confidence sampling, while random sampling also differs significantly from least-confidence sampling. Taken together, these results suggest that querying strategy has a *consistent but moderate impact* on SeqFAL's performance, with margin sampling providing a reliable yet incremental advantage over alternative strategies.

Table 3. Querying-strategy-level performance in SeqFAL. Results are averaged within each random seed and then aggregated over 10 independent runs. We report mean $\pm$ standard deviation and 95% confidence intervals for weighted F1-score ($wF1$), weighted precision (wP), and weighted recall (wR).

Querying Strategy	$wF1$ (Mean $\pm$ Std)	$wF1$ (95% CI)	wP (Mean $\pm$ Std)	wR (Mean $\pm$ Std)
FL-MarginAL (SeqFAL)	0.831 ± 0.031	[0.808, 0.853]	0.854 ± 0.021	0.846 ± 0.022
FL-RandomAL	0.811 ± 0.030	[0.789, 0.832]	0.841 ± 0.020	0.830 ± 0.021
FL-LeastConfidenceAL	0.795 ± 0.034	[0.771, 0.820]	0.833 ± 0.024	0.821 ± 0.025

Effect of Federation Size (K). Figure 3 presents the impact of federation size on performance, reporting both **accuracy** and **weighted F1-score** as the number of participating clients increases. Overall, both metrics improve when moving from smaller to larger federations, indicating that incorporating additional clients generally benefits collaborative learning in *SeqFAL*. Performance gains are most pronounced when increasing K from very small federation sizes, while improvements become less uniform at larger values of K .

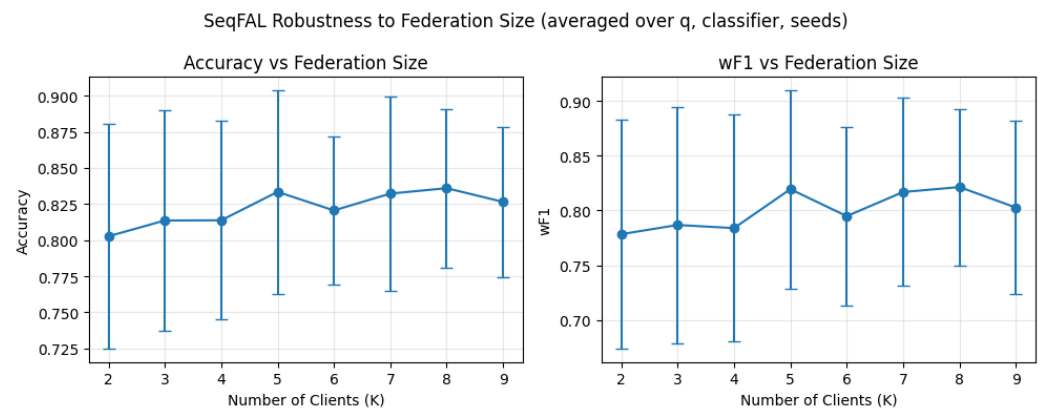


Figure 3. Accuracy and weighted F1-score as a function of federation size (K), averaged over experimental settings and random seeds.

To assess whether the observed differences across federation sizes are consistent across experimental contexts, we applied a Friedman test on weighted F1-score, treating K as the factor of interest and using q , rounds, classifier, and query as blocking variables. The test indicates a statistically significant effect of federation size ($\chi^2 = 614.92$, $p < 0.001$). The associated effect size, measured using Kendall's W , is $W = 0.488$, corresponding to a *moderate* effect. Post-hoc Nemenyi pairwise comparisons show that smaller federation sizes (e.g., $K = 2$ and $K = 3$) differ significantly from most larger configurations, whereas differences among medium-to-large federation sizes are less frequently significant. This pattern suggests that increasing the number of clients yields consistent improvements at early stages, while additional gains become more incremental as the federation grows.

Figure 4 further illustrates the interaction between federation size and the number of communication rounds. The heatmap shows that, under a limited communication budget, increasing K leads to only modest improvements in weighted F1-score. In contrast, when more communication rounds are available, larger federations consistently achieve higher performance. This interaction indicates that the apparent saturation observed at larger values of K is partly attributable to a fixed number of communication rounds rather than an inherent limitation of federation size. Overall, these results indicate that federation size has a systematic but moderate impact on *SeqFAL*'s performance. Larger federations are particularly beneficial when sufficient communication rounds are available, highlighting the importance of jointly considering federation scale and training horizon in federated active learning settings.

To better interpret the effect of federation size K observed in RQ1, we analyzed the class distribution heterogeneity induced by the Dirichlet-based client partitioning. Table A1 reports client-level non-IID statistics across different values of K , including entropy, Gini coefficient, and the fraction of single-class clients. Across all federation sizes ($K = 2 \dots 9$), these measures indicate moderate and stable non-IID conditions, with fewer than 9% of clients containing only a single class. Although increasing K leads to finer data fragmentation, it does not produce extreme or pathological client distributions. This controlled increase in heterogeneity explains why the Friedman test detects a statistically significant ef-

fect of federation size, while the corresponding effect size measured by Kendall's W remains moderate, resulting in small practical performance differences across federation sizes.

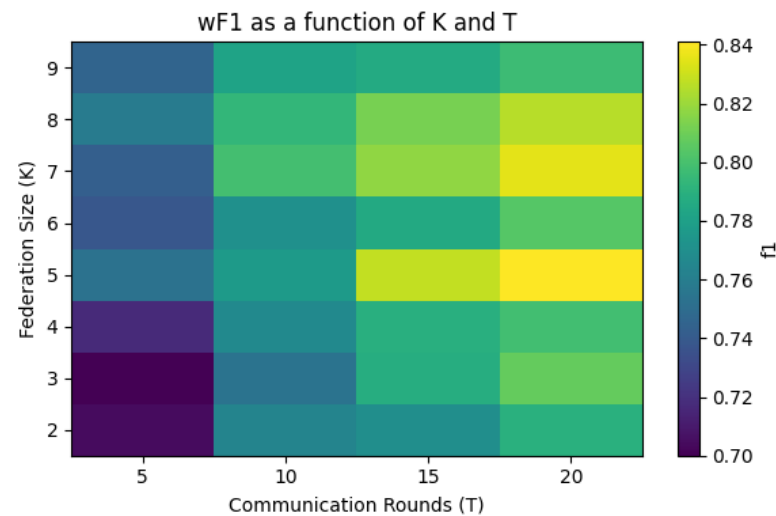


Figure 4. Interaction between federation size (K) and communication rounds (T) as a heatmap in terms of weighted F1-score.

We further relate these non-IID properties to the behavior of the fallback sampling mechanism. Fallback random sampling is triggered whenever a client-side querying strategy—margin-based, least-confidence, or random sampling—cannot be applied due to insufficient or degenerate local candidate pools. As shown in Table A2, fallback usage remains limited (approximately 15.8% overall) and does not increase monotonically with federation size, with rates ranging between 10% and 19%. This indicates that, despite realistic non-IID data distributions, client-side querying remains applicable in most cases. Together, these observations confirm that *SeqFAL*'s robustness to federation size arises from operating under controlled non-IID conditions and from the stability of its querying mechanisms, rather than from insensitivity to data heterogeneity.

Effect of Communication Rounds (T). Figure 5 illustrates the impact of the number of communication rounds on **accuracy** and **weighted F1-score**. Both metrics improve steadily as the number of rounds increases, indicating that additional communication enables more effective model refinement within *SeqFAL*. Performance gains are most pronounced when increasing the number of rounds from very small values, while improvements become more gradual at later stages, suggesting convergence behavior rather than instability.

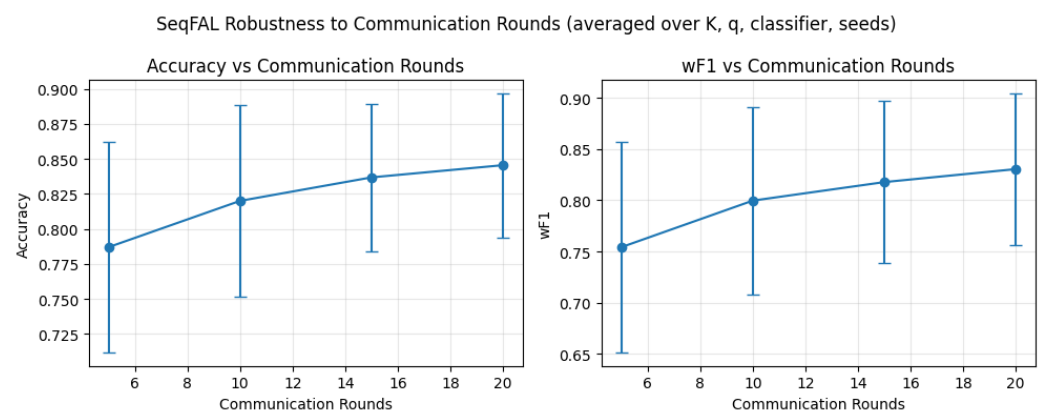


Figure 5. Accuracy and weighted F1-score as a function of the number of communication rounds (T), averaged over experimental settings and random seeds.

To evaluate whether the observed differences across communication rounds are consistent across experimental contexts, we applied a Friedman test on weighted F1-score, treating rounds as the factor of interest and using K , q , classifier, and query as blocking variables. The test indicates a statistically significant effect of the number of communication rounds ($\chi^2 = 852.51$, $p < 0.001$). The associated effect size, measured using Kendall's W , is $W = 0.789$, corresponding to a *large* effect. Post-hoc Nemenyi pairwise comparisons reveal statistically significant differences between all pairs of communication rounds, indicating that increasing the number of rounds consistently improves performance across blocks. In contrast to other factors discussed so far, the large effect size suggests that communication rounds constitute a dominant driver of performance within *SeqFAL*.

Overall, these results indicate that the number of communication rounds has a strong and systematic impact on *SeqFAL*'s performance. Increasing the communication budget yields substantial improvements in both accuracy and weighted F1-score, underscoring the importance of sufficient training rounds to fully exploit the benefits of federated active learning.

Effect of Query Size (q). Figure 6 shows the effect of the query size per communication round on **accuracy** and **weighted F1-score**. Both metrics increase monotonically as the query budget grows, indicating that acquiring more labeled instances per round consistently improves model performance within *SeqFAL*. The gains are most pronounced when moving from very small query sizes, while improvements become more gradual at higher values of q , suggesting diminishing marginal returns as the labeling budget increases.

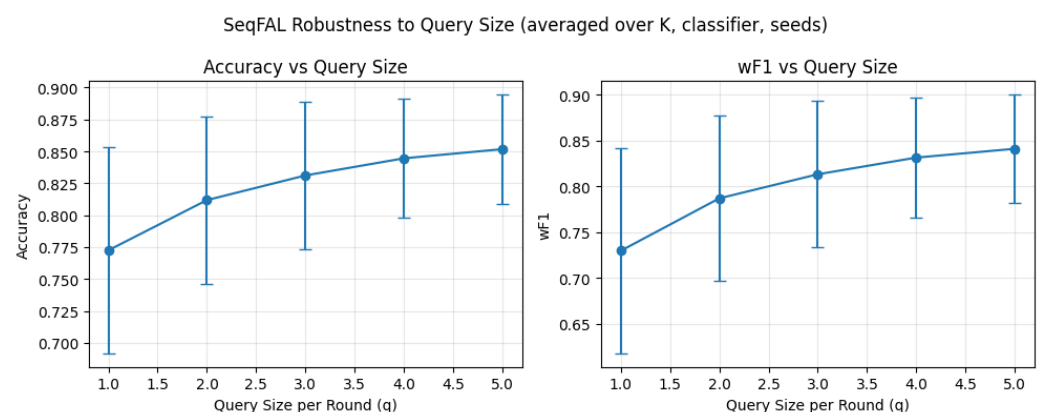


Figure 6. Effect of the query size per round (q) on performance in *SeqFAL*. Accuracy and $wF1$ are reported as a function of the labeling budget per communication round, averaged over federation sizes, communication rounds, classifiers, querying strategies, and random seeds.

To evaluate whether the observed differences across query sizes are consistent across experimental contexts, we applied a Friedman test on weighted F1-score, treating q as the factor of interest and using K , rounds, classifier, and query as blocking variables. The test indicates a statistically significant effect of the query size ($\chi^2 = 1072.16$, $p < 0.001$). The associated effect size, measured using Kendall's W , is $W = 0.931$, corresponding to a *large* effect. Post-hoc Nemenyi pairwise comparisons reveal statistically significant differences between nearly all query-size levels, indicating that increasing the number of queried instances per round yields consistent performance improvements across blocks. In contrast to other factors, the very large effect size highlights the dominant role of the query budget in shaping learning effectiveness within *SeqFAL*.

Figure 7 further illustrates the interaction between querying strategy and query size. When the query budget is small, margin sampling yields a higher weighted $wF1$ than both random and least-confidence sampling, indicating more effective use of limited labeling resources. As the query size increases, performance improves across all strategies and

the relative differences between them diminish. This interaction suggests that querying strategy plays a particularly important role under constrained labeling budgets, while the query size remains the dominant factor governing overall performance.

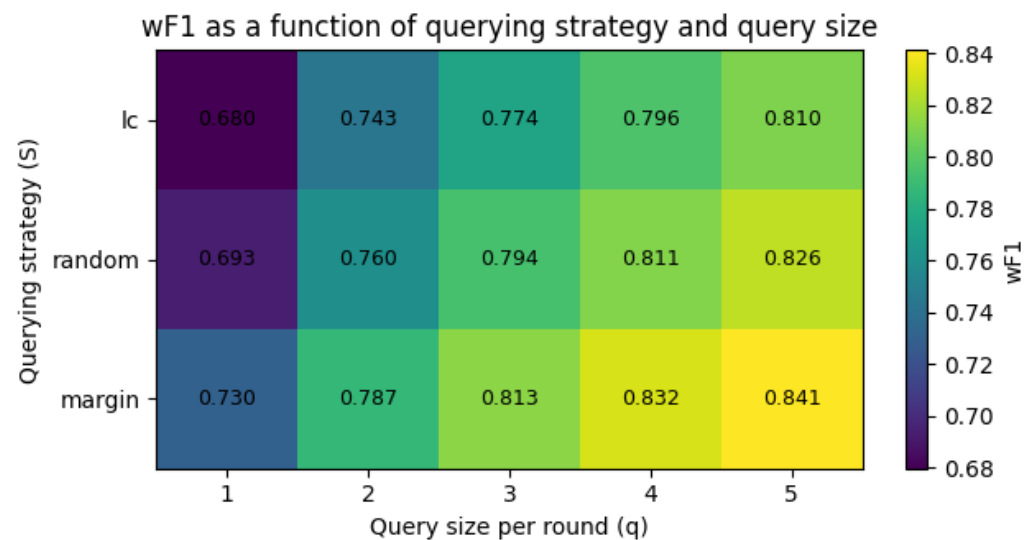


Figure 7. Interaction between querying strategy (S) and query size (q) in terms of weighted F1-score. The advantage of margin sampling is most pronounced at small query sizes, while differences across strategies diminish as the labeling budget increases. Results are averaged over federation sizes, communication rounds, classifiers, and random seeds.

Overall, these results indicate that the query size per round has a strong and systematic impact on *SeqFAL*'s performance. Larger query budgets substantially enhance both accuracy and weighted F1-score, emphasizing the importance of allocating sufficient labeling resources in federated active learning settings.

5.2. Answers to RQ2: *SeqFAL*'s Effectiveness Compared with Baselines

To assess the effectiveness of *SeqFAL*, we compare its performance with three baseline systems: centralized learning (Upper-Bound), active learning without federated aggregation (AL-NoFL), and federated learning without active querying (FL-NoAL).

Table 4 reports the absolute performance of all methods in terms of weighted F1-score ($wF1$), weighted precision (wP), and weighted recall (wR), while Table 5 summarizes the performance differences between each baseline and *SeqFAL* using mean $\pm$ standard deviation of the deltas across random seeds.

Table 4. Overall performance comparison between *SeqFAL* and baseline methods. Results are reported using mean $\pm$ standard deviation and 95% confidence intervals for weighted F1-score ($wF1$).

Method	Settings	$wF1$ (Mean $\pm$ Std)	$wF1$ (95% CI)
UpperBound	Centralized	0.902 ± 0.020	[0.888, 0.917]
AL-NoFL	Centralized	0.802 ± 0.029	[0.781, 0.822]
FL-NoAL	Distributed	0.555 ± 0.031	[0.533, 0.578]
<i>SeqFAL</i> (Ours)	Distributed	0.801 ± 0.029	[0.780, 0.821]

The centralized Upper-Bound achieves the highest overall performance, as expected, since it has full access to the entire labeled dataset and does not operate under federated or labeling constraints. Specifically, Upper-Bound attains a $wF1$ -score of 0.902 ± 0.020 , outperforming all distributed and label-efficient settings. However, *SeqFAL* achieves performance that closely approaches this upper bound while requiring substantially fewer

labeled samples and preserving data locality. The performance gap between *SeqFAL* and Upper-Bound is reflected by a positive delta of $+0.102 \pm 0.023$ in *wF1* and $+0.081 \pm 0.014$ in accuracy, indicating the remaining advantage of centralized access to labeled data.

Table 5. Performance differences between baseline methods and *SeqFAL*. Results are reported as mean $\pm$ standard deviation of the performance deltas across random seeds. Negative values indicate that *SeqFAL* outperforms the corresponding baseline.

Method	Settings	ΔAcc (Mean $\pm$ Std)	$\Delta wF1$ (Mean $\pm$ Std)
UpperBound	Centralized	$+0.081 \pm 0.014$	$+0.102 \pm 0.023$
AL-noFAL	Centralized	$+0.000 \pm 0.002$	$+0.001 \pm 0.002$
FL-NoAL	Distributed	-0.174 ± 0.048	-0.245 ± 0.054

When compared to AL-AL-NoFL, *SeqFAL* exhibits nearly identical performance across all reported metrics. Both methods achieve comparable *wF1*-scores (0.801 ± 0.029 for *SeqFAL* versus 0.802 ± 0.029 for AL-AL-NoFL), as well as matching weighted precision and recall values. This is further confirmed by the delta analysis in Table 5, where the performance differences between AL-AL-NoFL and *SeqFAL* are negligible ($\Delta wF1 = +0.001 \pm 0.002$, $\Delta\text{Acc} = +0.000 \pm 0.002$). These results indicate that, under frozen sentence embeddings and linear classifiers, the use of margin-based uncertainty sampling—originally proposed as part of *SeqFAL*—accounts for the majority of the observed performance gains, while iterative federated aggregation provides limited additional improvement beyond one-shot model fusion in this setting.

SeqFAL shows a substantial advantage over FL-NoAL, which lacks an active querying mechanism. FL-NoAL performs the weakest among the baselines, with an average *wF1*-score of 0.555 ± 0.031 . *SeqFAL* improves upon FL-NoAL by a large margin, as reflected by negative deltas of -0.245 ± 0.054 in *wF1* and -0.174 ± 0.048 in accuracy. This highlights that federated learning alone is insufficient for this task when clients start with minimal labeled data, and that active querying is essential for effective learning under label-scarce and non-IID conditions.

Overall, these results demonstrate that *SeqFAL* successfully bridges the gap between centralized learning and distributed, privacy-preserving training. While it cannot fully match the performance of a centralized oracle, *SeqFAL* preserves the full benefits of active learning and substantially outperforms federated learning without querying. The close alignment between *SeqFAL* and AL-AL-NoFL further suggests that, in the considered setting, label selection plays a more critical role than repeated parameter aggregation, providing valuable insight into the design trade-offs of federated active learning systems.

6. Discussion

6.1. Key Findings

The empirical evaluation of *SeqFAL* yields several important insights into the effectiveness, efficiency, and robustness of the proposed framework:

- ***SeqFAL* substantially outperforms FL-only baselines, while closely matching AL-only performance.** Across all weighted metrics, *SeqFAL* consistently outperforms Federated Learning without Active Learning (FL-NoAL), highlighting the critical role of uncertainty-based querying under label-scarce and non-IID conditions. In contrast, *SeqFAL* achieves performance nearly identical to Active Learning without Federated Learning (AL-NoFL) when both employ margin-based sampling, indicating that label selection is the dominant factor driving performance in this setting.

- **Near-centralized performance without data sharing.** *SeqFAL* approaches the performance of the centralized upper-bound model, despite operating under decentralized, label-scarce, and privacy-preserving constraints.
- **Margin-based active learning drives label efficiency.** Using margin-based uncertainty sampling—originally proposed as part of *SeqFAL*—substantially improves performance with limited labeled data. Larger query sizes consistently yield gains up to moderate budgets, beyond which improvements saturate, indicating an effective trade-off between annotation cost and predictive performance.
- **Communication rounds support convergence but exhibit diminishing returns.** Increasing the number of communication rounds improves performance and stability, with convergence typically observed around $T \geq 15$. However, once sufficient alignment is achieved, additional rounds provide diminishing benefits, particularly when models are retrained on frozen sentence embeddings.
- **Robustness to federation size.** Performance remains stable across a wide range of client counts ($K = 2 \dots 9$). While the Friedman test indicates a statistically significant effect of federation size, the associated effect size measured by Kendall's W is moderate, suggesting that the magnitude of this effect is limited in practice. Overall performance differences across federation sizes remain small, indicating that *SeqFAL* is largely robust to data fragmentation and heterogeneous local dataset sizes.
- **Efficient and privacy-preserving learning.** By exchanging only lightweight model parameters and using a limited number of labeled samples, *SeqFAL* reduces annotation cost and communication overhead while maintaining privacy across distributed stakeholders.
- **Statistical validation confirms practical relevance of design choices.** Non-parametric statistical tests, complemented by effect size analysis using Kendall's W , confirm that query size and communication rounds have statistically significant and practically meaningful effects on performance, whereas federation size exhibits a negligible effect. These findings support the scalability and robustness of *SeqFAL* in realistic distributed settings.

6.2. Practical Implications

From an applied perspective, the findings of this study demonstrate that organizations can collaboratively improve security requirements classification without sharing raw specification documents or sensitive textual artifacts, in line with the core premise of federated learning for privacy-preserving collaboration [10,12]. *SeqFAL*'s combination of federated learning and client-side active learning enables teams to benefit from cross-organizational knowledge while retaining full control over their local requirement repositories, which reflects how enterprises increasingly use FL to respect legal and competitive data-sharing constraints [37]. Because the framework relies on frozen sentence embeddings and lightweight linear classifiers, it imposes minimal computational and communication overhead, making it suitable for small teams, resource-constrained environments, and organizations without access to high-performance hardware [11,15].

In industrial practice, *SeqFAL* can be integrated into requirements management workflows to automatically flag security-related requirements during early development stages, even when contributors are distributed across departments, suppliers, or partner organizations. This aligns with the observation that FL is particularly attractive in multi-stakeholder, enterprise settings where data cannot be centrally pooled but joint models are still desirable [10,37]. Its reduced annotation cost—achieved through margin-based active learning—helps teams with limited labeling capacity maintain or improve classification quality, extending existing evidence that active learning can substantially lower labeling requirements in text classification and similar NLP tasks [17]. Moreover, the strict

locality of requirement text and the parameter-only aggregation strategy align well with confidentiality and compliance demands in domains such as finance, telecommunications, automotive, and critical infrastructure, where on-premise training and secure aggregation are recommended best practices [12,37].

Overall, *SeqFAL* underscores the practical value of combining privacy preservation, label efficiency, and computational simplicity when designing federated NLP systems for software engineering tasks. The results highlight the trade-offs that organizations must consider when selecting distributed learning strategies and point toward federated active learning as a promising approach for scalable and secure requirements engineering pipelines, consistent with broader calls for enterprise-ready, governance-aware FL deployments [12,37].

6.3. Limitations

This study operates within practical boundaries. Our experiments rely on controlled simulations of federated learning, and the findings naturally reflect this structured environment. We use established performance metrics, a margin-based sampling strategy, and statistical significance testing to support the reliability of comparisons, though these represent only a portion of the broader methodological space. Factors such as dataset scale and fixed annotation budgets also shape the variability observed across configurations. Even so, the approach is strengthened by extensive experimentation across multiple settings, transparent reporting, and openly shared code at <https://github.com/waadalhoshan/SeqFAL> (accessed on 29 November 2025), providing a solid and reproducible basis for evaluating *SeqFAL* in privacy-preserving, label-efficient requirements classification.

7. Conclusions

This paper introduced *SeqFAL*, a lightweight Federated Active Learning framework designed to support privacy-preserving and label-efficient classification of natural-language security requirements. Unlike existing approaches that rely on centralized datasets or computationally intensive deep learning models, *SeqFAL* combines frozen sentence embeddings, linear classifiers, and margin-based active learning within a federated setting, enabling organizations to collaboratively improve classification performance without sharing raw requirement text. The framework is designed to operate under realistic constraints observed in industrial environments, including limited labeled data, heterogeneous client distributions, and strict confidentiality requirements.

Our empirical evaluation on the SeqReq dataset demonstrates that *SeqFAL* consistently outperforms federated learning without active querying (FL-NoAL) and reaches comparable performance to centralized active learning without federation (AL-NoFL). Across a range of federation sizes, query budgets, and communication rounds, *SeqFAL* achieves performance fair-enough considering its federated settings to the centralized upper-bound while requiring substantially fewer labeled samples. Ablation studies further reveal that active learning plays a critical role in improving label efficiency, whereas the combination of client-side querying and global aggregation enables *SeqFAL* to remain robust even under non-IID data partitions.

These findings highlight the practical value of integrating active learning and federated learning for security-oriented requirements engineering tasks. *SeqFAL* offers a viable solution for organizations seeking to automate security requirement classification while maintaining data confidentiality and minimizing computational overhead. The framework's simplicity and low communication cost make it suitable for deployment in distributed software development environments, including small teams and resource-constrained settings.

Future work will explore alternative uncertainty-sampling strategies, investigate personalized federated variants, and examine how *SeqFAL* can be applied to additional forms of engineering documentation. Another promising direction is to complement the current evaluations with a lightweight security analysis of the framework itself to better understand its robustness against potential privacy or integrity risks in federated environments. For instance, extending *SeqFAL* to explicitly model adversarial clients and to evaluate robustness against model poisoning and inference attacks would strengthen its applicability to security-critical settings.

In addition, future work will investigate the use of more expressive, non-linear models and multi-class formulations, along with suitable uncertainty estimation strategies, to support scenarios involving complex decision boundaries and fine-grained requirement categories.

Finally, while this work evaluates *SeqFAL* on increasingly larger industrial datasets, further validation across additional domains and organizational contexts remains an important next step.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and data presented in the study are openly available in Github at <https://github.com/waadalhoshan/SeqFAL> (accessed on 29 November 2025).

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RE	Requirements Engineering
NLP	Natural Language Processing
NLU	Natural Language Understanding
FL	Federated Learning
AL	Active Learning

Appendix A

Table A1. Non-IID client data distribution statistics across federation sizes. Entropy and Gini coefficients are reported at the client level and averaged across clients for each federation size K .

K	#Clients	Single-Class Frac.	Entropy (Mean)	Entropy (Std)	Gini (Mean)	Gini (Std)	p_1 (Mean)
2	2	0.000000	0.475121	0.192279	0.313245	0.147408	0.321863
3	3	0.033333	0.480396	0.190714	0.317390	0.149256	0.294239
4	4	0.050000	0.412035	0.236457	0.270091	0.175775	0.363543
5	5	0.060000	0.431104	0.218243	0.281548	0.165529	0.379986
6	6	0.066667	0.445109	0.231972	0.295432	0.176326	0.333035
7	7	0.085714	0.424090	0.215181	0.275675	0.160947	0.369591
8	8	0.050000	0.466267	0.189026	0.305348	0.146006	0.378713
9	9	0.055556	0.445662	0.217613	0.293118	0.167078	0.374686

Table A2. Fallback random sampling usage across federation sizes. Fallback is triggered when a client-side querying strategy (margin-based, least-confidence, or random sampling) cannot select a valid sample due to local data constraints.

K	Fallback Rate	Client-Rounds	Fallback Rate (%)
2	0.104630	48,600	10.46
3	0.126543	72,900	12.65
4	0.190741	97,200	19.07

Table A2. Cont.

K	Fallback Rate	Client-Rounds	Fallback Rate (%)
5	0.158889	121,500	15.89
6	0.152469	145,800	15.25
7	0.181217	170,100	18.12
8	0.142824	194,400	14.28
9	0.163169	218,700	16.32

References

- Zhao, L.; Alhoshan, W.; Ferrari, A.; Letsholo, K.J.; Ajagbe, M.A.; Chioasca, E.V.; Batista-Navarro, R.T. Natural language processing for requirements engineering: A systematic mapping study. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–41. [\[CrossRef\]](#)
- Knauss, E.; Houmb, S.; Schneider, K.; Islam, S.; Jürjens, J. Supporting requirements engineers in recognising security issues. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality, (REFSQ 2011), Essen, Germany, 28–30 March 2011; pp. 4–18. [\[CrossRef\]](#)
- Firesmith, D. Engineering security requirements. *J. Object Technol.* **2003**, *2*, 53–68. [\[CrossRef\]](#)
- Kadebu, P.; Sikka, S.; Tyagi, R.K.; Chiurunge, P. A classification approach for software requirements towards maintainable security. *Sci. Afr.* **2023**, *19*, e01496. [\[CrossRef\]](#)
- Varenov, V.; Gabdrahmanov, A. Security requirements classification into groups using nlp transformers. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), Notre Dame, IN, USA, 20–24 September 2021; pp. 444–450. [\[CrossRef\]](#)
- Batool, R.; Naseer, A.; Maqbool, A.; Kayani, M. Automated categorization of software security requirements: An NLP and ML based approach. *Requir. Eng.* **2025**, *30*, 133–145. [\[CrossRef\]](#)
- Zhao, L.; Alhoshan, W. Machine Learning for Requirements Classification. In *Handbook on Natural Language Processing for Requirements Engineering*; Springer: Cham, Switzerland, 2025; pp. 19–59. [\[CrossRef\]](#)
- Haley, C.; Laney, R.; Moffett, J.; Nuseibeh, B. Security Requirements Engineering: A Framework for Representation and Analysis. *IEEE Trans. Softw. Eng.* **2008**, *34*, 133–153. [\[CrossRef\]](#)
- Khan, R.A.; Khan, S.U.; Khan, H.U.; Ilyas, M. Systematic Mapping Study on Security Approaches in Secure Software Engineering. *IEEE Access* **2021**, *9*, 19139–19160. [\[CrossRef\]](#)
- Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [\[CrossRef\]](#)
- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282. Available online: <https://proceedings.mlr.press/v54/mcmahan17a.html> (accessed on 25 November 2025).
- Kairouz, P.; McMahan, H.B. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [\[CrossRef\]](#)
- Nasajpour, M.; Pouriyeh, S.; Parizi, R.M.; Han, M.; Mosaiyebzadeh, F.; Liu, L.; Xie, Y.; Batista, D.M. Federated Learning in Smart Healthcare: A Survey of Applications, Challenges, and Future Directions. *Electronics* **2025**, *14*, 1750. [\[CrossRef\]](#)
- Kumar, R.; Shieh, C.S.; Chakrabarti, P.; Kumar, A.; Moolchandani, J.; Sinha, R. Privacy-Preserving Federated Learning in Healthcare, E-Commerce, and Finance: A Taxonomy of Security Threats and Mitigation Strategies. *EPJ Web Conf.* **2025**, *328*, 01066. [\[CrossRef\]](#)
- Asad, M.; Otoum, S.; Shaukat, S. Clients Eligibility-Based Lightweight Protocol in Federated Learning: An IDS Use Case. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 3759–3774. [\[CrossRef\]](#)
- Wang, R.; Liu, J.; Zhang, Q.; Fu, C.; hou, Y. Federated learning for feature-fusion based requirement classification. *Clust. Comput.* **2024**, *27*, 3397–3416. [\[CrossRef\]](#)
- Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report; University of Wisconsin–Madison: Madison, WI, USA, 2009; Volume 1648. Available online: <http://digital.library.wisc.edu/1793/60660> (accessed on 25 November 2025).
- Li, D.; Wang, Z.; Chen, Y.; Jiang, R.; Ding, W.; Okumura, M. A Survey on Deep Active Learning: Recent Advances and New Frontiers. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 5879–5899. [\[CrossRef\]](#)
- Shelmanov, A.; Puzyrev, D.; Kupriyanova, L.; Belyakov, D.; Larionov, D.; Khromov, N.; Kozlova, O.; Artemova, E.; Dylov, D.V.; Panchenko, A. Active Learning for Sequence Tagging with Deep Pre-trained Models and Bayesian Uncertainty Estimates. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 1698–1712. [\[CrossRef\]](#)

20. Magalhães, C.; Araujo, J.; Sardinha, A. MARE: An Active Learning Approach for Requirements Classification. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, 20–24 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 222–233. [\[CrossRef\]](#)
21. Riesener, M.; Kuhn, M.; Schümmelfeder, S.; Xiao, D.; Norheim, J.; Rebentisch, E.; Schuh, G. Active Learning with Pre-trained Language Models for Named Entity Recognition in Requirements Engineering. *Procedia CIRP* **2024**, *128*, 339–344. [\[CrossRef\]](#)
22. Hassani, S.; Sabetzadeh, M.; Amyot, D. An empirical study on LLM-based classification of requirements-related provisions in food-safety regulations. *Empir. Softw. Eng.* **2025**, *30*, 72. [\[CrossRef\]](#)
23. Alhoshan, W.; Ferrari, A.; Zhao, L. Zero-shot learning for requirements classification: An exploratory study. *Inf. Softw. Technol.* **2023**, *159*, 107202. [\[CrossRef\]](#)
24. Petrillo, L.; Martinelli, F.; Santone, A.; Mercaldo, F. Explainable Security Requirements Classification Through Transformer Models. *Future Internet* **2025**, *17*, 15. [\[CrossRef\]](#)
25. Cleland-Huang, J.; Mazrouee, S.; Liguio, H.; Port, D. *NFR Dataset*; Zenodo: Geneva, Switzerland, 2007. [\[CrossRef\]](#)
26. Sun, Y.; Pan, S.; Sun, A.; Fu, Z.; Long, S.; Li, Z. FedLFP: Communication-Efficient Personalized Federated Learning on Non-IID Data in Mobile Edge Computing Environments. *IEEE Trans. Mob. Comput.* **2025**, *24*, 8811–8823. [\[CrossRef\]](#)
27. Wang, J.; Zhang, S.; Xu, Y.; Zhang, Z.; Yang, X.; Cheng, Y. ILAMP: Improved text extraction from gradients in federated learning using language model priors and sequence beam search. *Expert Syst. Appl.* **2025**, *271*, 126592. [\[CrossRef\]](#)
28. Wang, R.; Yu, T.; Zhang, R.; Kim, S.; Rossi, R.; Zhao, H.; Wu, J.; Mitra, S.; Yao, L.; Henao, R. Personalized Federated Learning for Text Classification with Gradient-Free Prompt Tuning. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, 16–21 June 2024; pp. 4597–4612. [\[CrossRef\]](#)
29. Huo, Z.; Fan, Y.; Huang, Y. A Communication-Efficient Federated Text Classification Method Based on Parameter Pruning. *Mathematics* **2023**, *11*, 2804. [\[CrossRef\]](#)
30. Casillo, F.; Deufemia, V.; Gravino, C. Beyond domain dependency in security requirements identification. *Inf. Softw. Technol.* **2025**, *182*, 107702. [\[CrossRef\]](#)
31. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [\[CrossRef\]](#)
32. Knauss, E.; Houmb, S.H.; Islam, S.; Jürjens, J.; Schneider, K. *SecReq*; Zenodo: Geneva, Switzerland, 2021. [\[CrossRef\]](#)
33. Kobilica, A.; Ayub, M.; Hassine, J. Automated identification of security requirements: A machine learning approach. In Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering, Trondheim, Norway, 15–17 April 2020; pp. 475–480.
34. Cleland-Huang, J.; Settimi, R.; Zou, X.; Solc, P. Automated classification of non-functional requirements. *Requir. Eng.* **2007**, *12*, 103–120. [\[CrossRef\]](#)
35. Zadenoori, M.A.; Zhao, L.; Alhoshan, W.; Ferrari, A. Automatic Prompt Engineering: The Case of Requirements Classification. In *Requirements Engineering: Foundation for Software Quality, 31st International Working Conference, REFSQ 2025, Barcelona, Spain, 7–10 April 2025, Proceedings*; Hess, A., Susi, A., Eds.; Springer: Cham, Switzerland, 2025; pp. 217–225.
36. Alhoshan, W.; Zhao, L.; Ferrari, A.; Letsholo, K.J. A Zero-Shot Learning Approach to Classifying Requirements: A Preliminary Study. In *Requirements Engineering: Foundation for Software Quality: 28th International Working Conference, REFSQ 2022, Birmingham, UK, 21–24 March 2022, Proceedings*; Gervasi, V., Vogelsang, A., Eds.; Springer: Cham, Switzerland, 2022; pp. 52–59.
37. Hasan, M.M. Federated Learning Models for Privacy-Preserving AI In Enterprise Decision Systems. *Int. J. Bus. Econ. Insights* **2025**, *5*, 238–269. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.