Federated Learning for Document Classification

***Abstract*—Document classification is a fundamental task in the realm of natural language processing (NLP) that involves categorizing documents into predefined classes based on their content. Machine learning models have become an integral part of document classification to enable the automatic analysis and categorization of textual data based on their content. A typical paradigm is to collect labeled documents from different remote users/devices, train the machine learning model on the server, and deploy the trained model on different devices serving different users. However, this centralized training paradigm would cause privacy concerns due to the leakage of sensitive information contained in the documents shared with the server. We leverage federated learning for document classification to maintain higher utility performance by training the model using all labeled documents while avoiding violating privacy concerns. Specifically, we maintain a global model on the centralized server and set up three local models on the remote devices, each of which is equipped with its own subset of documents. During each training epoch, local models first receive the aggregated parameters from the global model and then update their parameters by performing gradient descent when optimizing the document classification loss over their own subset of data. The updated parameters are then returned to the global model for parameter aggregation. We framework the message-passing between the global and local models within the PUB-SUB mechanism. Experimental results demonstrate that our federated learning framework successfully protects user privacy while achieving the same performance as centralized training.**

***Index Terms*—Document classification, privacy concern, federated learning, PUB-SUB framework**

I. Introduction

Document classification is a critical task in natural language processing (NLP) that involves categorizing text documents into predefined classes or categories based on their content [1]–[3]. This process enables the efficient organization, management, and retrieval of information from vast text data repositories, transforming how we interact with digital content [4], [5]. By leveraging advanced machine learning algorithms and techniques, document classification systems can automatically analyze, understand, and classify text data, streamlining information filtering and retrieval processes while reducing human intervention and error [6], [7]. Document classification applications are vast and diverse, spanning various domains such as spam detection [8], sentiment analysis [9], topic identification [10], document tagging [11], content recommendation [12], and automated customer support [13]. As the volume of digital text data continues to grow exponentially, document classification plays an increasingly vital role in the world of information technology, making it essential for businesses and organizations to stay ahead in a data-driven landscape.

The authors are with the Department of Computer Science, Vanderbilt University, Nashville, TN 37212, USA (email: yu.wang.1@vanderbilt.edu; jiakai.long@vanderbilt.edu; nitish.r.nimma@vanderbilt.edu)

To guarantee high-performance of machine learning models for document classification, a common approach involves collecting a large number of labeled documents from different users or devices, training the model in a central server, and deploying it in different devices serving different users [14], [15]. This method helps the model generalize better across various data distributions and text types, ensuring a robust and effective classification system. However, this centralized approach may raise privacy concerns since sensitive information contained in the documents is implicitly shared from personal devices to the central server, which exposes users to potential data breaches, unauthorized access, or misuse of their private data [16]–[18]. For example, document classification could be used in healthcare to categorize electronic health records, research papers, or patient reports [19], [20]. Centralized storage and processing of such sensitive medical data could make it vulnerable to cyber-attacks, resulting in unauthorized access to confidential patient information. A notable case is the Anthem data breach in 2015, where hackers stole the personal information and medical records of nearly 78.8 million customers [21], [22]. Document classification models can be employed in the financial sector for tasks such as credit risk assessment [23], fraud detection [24], or categorizing loan applications [25]. Sharing sensitive financial documents with a central server may expose users' financial history, account numbers, or social security numbers to potential breaches. The Equifax data breach in 2017 is a real-life example of this type of risk, where the personal information of 147 million people, including their credit histories, was compromised [26].

Alternative training methodologies like federated learning [27] are being explored to address the above privacy concerns. Federated learning allows multiple devices to train a global model collaboratively without sharing the raw data, preserving data privacy while enabling effective model development [28]. By exchanging only model updates or gradients, federated learning minimizes the risks associated with centralized data storage and processing [29], ensuring that sensitive information remains secure on local devices. As data privacy needs continue to grow, incorporating privacy-preserving techniques like federated learning into document classification systems will become increasingly important to maintain user trust and comply with data protection regulations.

In this work, we expect to leverage federated learning on document classification to maintain higher performance while avoiding violating privacy regulations. Specifically, we maintain a global model on the centralized server and set up three local models on the remote devices, each of which is equipped with its own subset of documents. During each training epoch, local models first receive the aggregated parameters from the global model and then update their parameters by performing gradient descent when optimizing the document classification

loss over their own subset of data. The updated parameters are then returned to the global model for parameter aggregation. We model the message-passing between the global and local models as the PUB-SUB framework. For each of the local models it maintains one REQ and one SUB socket for each of the local models since it uses the REC socket to request the updated parameters from the global model and uses the SUB socket to listen to the global model for any refreshed parameters after global aggregation. For the global model, it maintains one REQ socket to respond to the local model's request and one PUB socket for publishing the aggregated parameters. The PUB-SUB framework enables efficient parameter sharing between local and global models, which mimics the training over the entire dataset while following the principle that personal data never leaves its own device. To summarize, our main contributions are as follows:

- To achieve a better trade-off between utility and privacy, we implement a PUB-SUB-based federated learning framework for document classification. The global model, as a Publisher, will keep publishing its aggregated parameters, and the local model, as a subscriber, will update their own parameters via gradient descent after receiving parameters from the global model.
- We collect a Wikipedia document dataset, which includes thousands of documents with their features being the bag-of-words of their summarization.
- We perform comprehensive experiments to demonstrate that our framework achieves higher utility performance while protecting user privacy.

II. Related work

A. Document Classification

Document classification, also known as text categorization or document categorization, is a fundamental task in natural language processing (NLP) and machine learning that involves automatically assigning predefined categories or labels to a given text document. This process enables efficient organization, retrieval, and management of large volumes of unstructured text data, such as emails, news articles, social media posts, and customer reviews. Document classification techniques can be broadly divided into supervised and unsupervised. Supervised techniques rely on pre-labeled training data, where documents are annotated with their corresponding categories. Popular supervised machine learning algorithms include Naive Bayes, Support Vector Machines, and Deep Learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). On the other hand, unsupervised techniques do not require labeled data. Instead, they rely on clustering or topic modeling algorithms, such as K-means, Latent Dirichlet Allocation (LDA), and Non-negative Matrix Factorization (NMF), to group similar documents together based on their content. Document classification applications span various domains, including sentiment analysis, spam detection, topic labeling, and automated tagging for content management systems. This work follows the supervised setting and trains our model on the labeled documents. Since we only aim to demonstrate the feasibility of leveraging federated learning in document classification, we choose the MLP-based model as our machine learning backbone.

B. Privacy concerns in document classification

Privacy issues in document classification arise when the process of training, deploying, or using classification models potentially exposes sensitive information about the individuals or entities that contributed to the dataset. This can lead to breaches of confidentiality, unauthorized access, or misuse of personal data, which can result in legal, ethical, and social implications. Key privacy concerns in document classification include:

Data privacy: Text documents used for classification may contain personally identifiable information (PII) or sensitive attributes, which must be protected to ensure compliance with data protection regulations, such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA).

Model inversion attacks: In a model inversion attack, an adversary can infer sensitive information about a specific individual by querying a trained classification model. For instance, an attacker may be able to reconstruct parts of the original text document, revealing sensitive data.

Membership inference attacks: These attacks aim to determine whether a specific data point was part of the training set used to build a classification model. If successful, an attacker could infer sensitive information about an individual or deduce that they belong to a specific group or category.

Attribute inference attacks: In these attacks, adversaries use the model's output to infer sensitive attributes of individuals, even if the model was not explicitly trained to predict these attributes.

C. Federated Learning

Federated learning is a decentralized approach to training machine learning models, which enables multiple devices or nodes to collaboratively learn from their local data while maintaining data privacy. Instead of sending raw data to a central server, the learning process occurs on each device, and only the model updates (i.e., gradients or weights) are shared with the central server. The server then aggregates these updates and disseminates the improved model back to the devices. This approach has several advantages over traditional centralized learning. From the data privacy perspective, since raw data never leaves the local devices, federated learning inherently preserves the privacy of the users. It reduces the risk of data breaches or misuse. From the perspective of reduced communication overhead, sharing model updates, which are typically smaller in size compared to raw data, reduces the bandwidth requirements and latency associated with data transmission. Better utilization of local resources: By leveraging the computational power of individual devices, federated learning can efficiently handle large-scale datasets without the need for centralized, high-performance computing resources. Federated learning is particularly well-suited for applications involving sensitive data, such as healthcare, finance, or mobile devices, where data privacy and security are of utmost

importance. However, it also presents challenges, including non-IID (Independent and Identically Distributed) data, device heterogeneity, and stragglers. These require novel optimization techniques and communication strategies to achieve robust and efficient learning.

III. Framework

A. Problem Definition

Let $\mathcal{D} = \{D_i\}_{i=1}^n = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^n$ denote the set of n documents, and D_i is the i^{th} document with its textual-based feature $\mathbf{X}_i$ and the label $\mathbf{Y}_i \in \{0, 1\}$ indicating whether the topic of the document is around a specific topic. We aim to learn document representations $\mathbf{H} \in \mathbb{R}^{n \times d'}$ with $\mathbf{H}_i$ for each $D_i \in \mathcal{D}$ that is well-predictive of its one-hot binary encoded label $\mathbf{Y}_i$. The problem of document classification can be formalized as follows:

Problem 1. *Given a set of attributed documents $\mathcal{D}$ with a subset of labeled documents $\mathcal{D}^\ell$, we aim to learn a document encoder and classifier $\mathcal{F} : \mathcal{F}(\mathbf{X}^{D_i}) \rightarrow \mathbf{Y}_i$ that works well for predicting document topics.*

B. Document Data Collection

In our document classification, each document D_i corresponds to a Wikipedia page historically viewed by a user, and we collect this data from here. For each document D_i, we call the Wikipedia API and implement a multi-parallel querying method to enable the fast query of the main page information of each Wikipedia article. An example of one document is shown in Fig 1 describing 'Taobao', which is one of the biggest Chinese e-commerce platforms. As our main purpose is to demonstrate the feasibility of applying federated learning in document classification to maintain utility performance and avoid raising privacy concerns, we expect to avoid using complex NLP-based models while only planning to use a simple MLP-based encoder. Therefore, we pre-process the textual-based description of each Wikipedia page by transforming them into a feature vector through sentence-transformer [30] and hence obtaining feature vectors $\mathbf{E} \in \mathbb{R}^{n \times d}$ for all n documents.

Fig. 1: An example of the document 'Taobao' Wikipedia page

To obtain the label/topic of each document, we follow [31] and classify each document into 64 binary labels based on whether their content involves the corresponding topic. Specifically, we select 'Technology' as the label, and the problem is to detect whether a given document talks about Technology.

C. System Architecture

As shown in Figure 2, we implement the whole system following the PUB-SUB framework. The system is hosted on Chameleon Cloud under an open-stack based server. Each node is configured with M1.medium option and are under the same network to ensure connectivity. Assuming without loss of generalizability, we have three worker nodes representing three remote devices/users and one master node representing the central server. During each training epoch, local models first receive the aggregated parameters from the global model via their SUB socket and then update their parameters by performing gradient descent when optimizing the document classification loss over their own subset of data. The updated parameters are then returned to the global model via a REQ-REP connection for parameter aggregation. We framework the message-passing between the global and local models within the PUB-SUB mechanism with the additional REQ-REP capability. The PUB-SUB sockets are used for network-wide parameter update broadcast, while the REQ-REP sockets are utilized when the worker nodes is reporting an updated results of training.

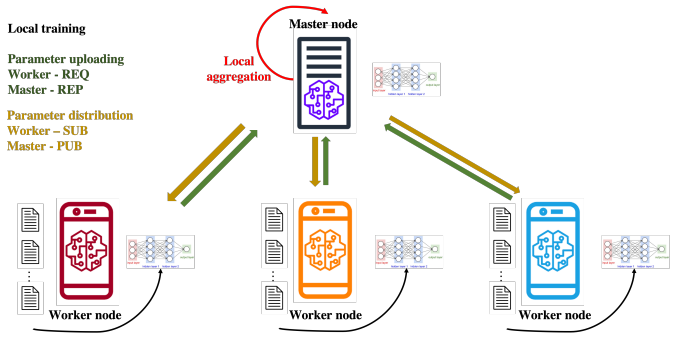

Fig. 2: The system architecture of our PUB-SUB-based federated learning framework for document classification

D. Machine Learning Model

Since we aim to demonstrate the feasibility of federated learning in document classification, we select the 2-layer multi-layer perceptron as our model backbone and maintain the same model architecture on all these four nodes to enable parameter sharing. During each training epoch, the master node will first receive the updated parameters from each worker node in the last epoch and then aggregate them together through mean pooling:

$$\mathbf{\Theta}_{\text{master}}^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{\Theta}_{\text{worker}_i}^{t} \tag{1}$$

After that, the master node sends the aggregated parameters back to the worker node. After receiving the aggregated parameters from the master node, each worker node then performs the gradient descent by optimizing the cross-entropy-loss on the document classification:

$$\mathbf{\Theta}_{\text{worker}_i}^{t+1} = \mathbf{\Theta}_{\text{master}_i}^{t+1} - \nabla_{\mathbf{\Theta}_{\text{worker}_i^t}} \mathcal{L}^{\text{ce}} \tag{2}$$

$$\mathcal{L}^{\text{ce}} = \mathbb{E}_{D_i \sim \mathcal{D}} - (y_i \log f(\mathbf{X}_i) + (1 - y_i) \log(1 - f(\mathbf{X}_i))) \tag{3}$$

Through the above forward-backward process, the whole model would be updated towards behaving well on classifying all documents from three worker nodes. Meanwhile, the documents never leave their corresponding device, hence protecting the user's privacy. We divide the above message-passing procedure into two directions: one is from the master node to the worker node, realizing the parameter distribution, and the other one is parameter uploading, realizing the model aggregation. In the parameter distribution stage, the worker node serves as the subscriber. It hence maintains a SUB socket, and the master node serves as the publisher and hence maintains a PUB socket. In the parameter uploading stage, the worker node serves as the requester and hence maintains a REQ socket, while the master node serves as the response and hence maintains a REP socket.

IV. Experiment

A. Performance Comparison

Here we compare the document classification performance on three devices between the conventional centralized training and the proposed decentralized/federated learning. We only use the training data stored on our own devices for centralized training. As shown in Figure 3, the distributed/federated training achieves higher performance than centralized training because the global aggregation procedure enables the parameters sharing among multiple devices and indirectly enhances the model generalizability. Moreover, comparing the performance gap among these three devices under these two settings, we find that distributed learning achieves higher performance gain on devices 2 and 3 than on 1. We hypothesize that the training signals provided by the data on devices 2 and 3 are not as poor as the one on Device 1, so borrowing information by parameter sharing from other devices leads to larger performance improvement.

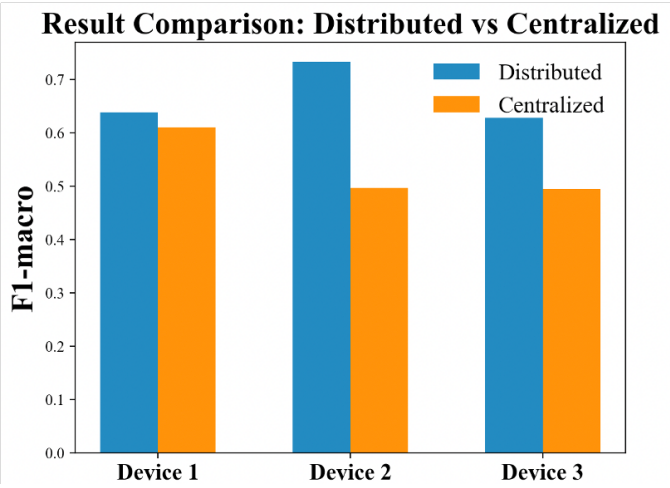

Fig. 3: The performance comparison between distributed training and centralized training

B. Efficiency Comparison

We also compare the efficiency of the whole framework by comparing the training time among multiple devices. As shown in Table **??**, our federated learning takes significantly longer time than centralized learning in training the model. This is because of the latency of the message-passing among multiple devices. However, in the real-world scenario, when we have a large number of documents on each worker node to train, the training time will increase significantly so that the time for message-passing would become less important. In that case, our model shares the same complexity as the non-federated learning one.

Scenario	Total Time (s)
Centralized Learning	6.56/4.51/6.33
Federated Learning	80.74

V. Conclusion

In order to maintain a high-level document classification performance while protecting users' privacy, we propose to use federated learning in document classification. Specifically, we implement a master node and three worker nodes. The master node is responsible for parameter aggregation and the worker node is responsible for parameter updating. The parameter sharing between the master node and worker nodes is realized by PUB-SUB message-passing mechanism. We collect the document data from Wikipedia and obtain its features/labels through a sentence transformer. Experimental results demonstrate that using our designed federated learning framework achieves better performance in training the model separately on three worker nodes. However, the time consumption of the whole federated learning process also increases due to message-passing. Future research directions include temporarily updating model parameters to achieve a better performance-efficiency trade-off.

References

[1] F. ul Hassan, T. Le, and D.-H. Tran, "Multi-class categorization of design-build contract requirements using text mining and natural language processing techniques," in *Construction Research Congress 2020: Project Management and Controls, Materials, and Contracts*. American Society of Civil Engineers Reston, VA, 2020, pp. 1266–1274.

[2] C. Wu, X. Li, Y. Guo, J. Wang, Z. Ren, M. Wang, and Z. Yang, "Natural language processing for smart construction: Current status and future directions," *Automation in Construction*, vol. 134, p. 104059, 2022.

[3] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[4] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International journal of information management*, vol. 35, no. 2, pp. 137–144, 2015.

[5] H. R. Nemati, D. M. Steiger, L. S. Iyer, and R. T. Herschel, "Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing," *Decision Support Systems*, vol. 33, no. 2, pp. 143–161, 2002.

[6] A. Y. Sun and B. R. Scanlon, "How can big data and machine learning benefit environment and water management: a survey of methods, applications, and future directions," *Environmental Research Letters*, vol. 14, no. 7, p. 073001, 2019.

[7] M. Thangaraj and M. Sivakami, "Text classification techniques: A literature review," *Interdisciplinary journal of information, knowledge, and management*, vol. 13, p. 117, 2018.

[8] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 1189–1190.

[9] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[10] V. Stoyanov and C. Cardie, "Topic identification for fine-grained opinion analysis," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, pp. 817–824.

[11] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He, "Document recommendation in social tagging services," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 391–400.

[12] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," *The adaptive web: methods and strategies of web personalization*, pp. 325–341, 2007.

[13] T. H. Davenport and P. Klahr, "Managing customer support knowledge," *California management review*, vol. 40, no. 3, pp. 195–208, 1998.

[14] X. Zhu and S. Gauch, "Incorporating quality metrics in centralized/distributed information retrieval on the world wide web," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, pp. 288–295.

[15] J. Arguello, J. Callan, and F. Diaz, "Classification-based resource selection," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1277–1286.

[16] K. Thomas, C. Grier, and D. M. Nicol, "unfriendly: Multi-party privacy risks in social networks," in *Privacy Enhancing Technologies: 10th International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings 10*. Springer, 2010, pp. 236–252.

[17] H. Mao, X. Shuai, and A. Kapadia, "Loose tweets: an analysis of privacy leaks on twitter," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, 2011, pp. 1–12.

[18] K. Häyrinen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: a review of the research literature," *International journal of medical informatics*, vol. 77, no. 5, pp. 291–304, 2008.

[19] T. A. Koleck, C. Dreisbach, P. E. Bourne, and S. Bakken, "Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review," *Journal of the American Medical Informatics Association*, vol. 26, no. 4, pp. 364–379, 2019.

[20] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore, "Automatic de-identification of textual documents in the electronic health record: a review of recent research," *BMC medical research methodology*, vol. 10, no. 1, pp. 1–16, 2010.

[21] L. H. Yeo and J. Banfield, "Human factors in electronic health records cybersecurity breach: an exploratory analysis," *Perspectives in Health Information Management*, vol. 19, no. Spring, 2022.

[22] K. T. Smith, A. Jones, L. Johnson, and L. M. Smith, "Examination of cybercrime and its effects on corporate stock value," *Journal of Information, Communication and Ethics in Society*, vol. 17, no. 1, pp. 42–60, 2019.

[23] N. Chen, B. Ribeiro, and A. Chen, "Financial credit risk assessment: a recent review," *Artificial Intelligence Review*, vol. 45, pp. 1–23, 2016.

[24] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical science*, vol. 17, no. 3, pp. 235–255, 2002.

[25] S.-T. Li, W. Shiue, and M.-H. Huang, "The evaluation of consumer loans using support vector machines," *Expert Systems with Applications*, vol. 30, no. 4, pp. 772–782, 2006.

[26] L. I. Labrecque, E. Markos, K. Swani, and P. Peña, "When data security goes wrong: Examining the impact of stress, social contract violation, and data type on consumer coping responses following a data breach," *Journal of Business Research*, vol. 135, pp. 559–571, 2021.

[27] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[28] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[29] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[30] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[31] I. Johnson, M. Gerlach, and D. Sáez-Trumper, "Language-agnostic topic classification for wikipedia," in *Companion Proceedings of the Web Conference 2021*, 2021, pp. 594–601.