Artificial Intelligence Review (2024) 57:320
https://doi.org/10.1007/s10462-024-10970-5

Federated learning-based natural language processing: a systematic literature review

Younas Khan[1] · **David Sánchez**[1] · **Josep Domingo-Ferrer**[1]

Accepted: 17 September 2024 / Published online: 12 October 2024
© The Author(s) 2024

Abstract
Federated learning (FL) is a decentralized machine learning (ML) framework that allows models to be trained without sharing the participants' local data. FL thus preserves privacy better than centralized machine learning. Since textual data (such as clinical records, posts in social networks, or search queries) often contain personal information, many natural language processing (NLP) tasks dealing with such data have shifted from the centralized to the FL setting. However, FL is not free from issues, including convergence and security vulnerabilities (due to unreliable or poisoned data introduced into the model), communication and computation bottlenecks, and even privacy attacks orchestrated by honest-but-curious servers. In this paper, we present a systematic literature review (SLR) of NLP applications in FL with a special focus on FL issues and the solutions proposed so far. Our review surveys 36 recent papers published in relevant venues, which are systematically analyzed and compared from multiple perspectives. As a result of the survey, we also identify the most outstanding challenges in the area.

Keywords Federated learning · Natural language processing · Privacy · Security · Systematic literature review

✉ Younas Khan
younas.khan@urv.cat

David Sánchez
david.sanchez@urv.cat

Josep Domingo-Ferrer
josep.domingo@urv.cat

[1] Department of Computer Engineering and Mathematics, CYBERCAT-Center for Cybersecurity Research of Catalonia, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Catalonia, Spain

1 Introduction

Data-driven machine learning (ML) models use iterative, intelligent algorithms to learn from features and patterns in data. With the availability of rich data sources, ML has seen rapid growth and development, and it has become an essential technology for today's world, as it can help tackle complex, high-dimensional challenges. Applications of ML can be found in various fields, including education, healthcare, stock exchange, banking, finance, marketing, social networks, and network security (Injadat et al. 2021).

However, the collection and management of data, particularly personal data, in traditional centralized ML settings often clash with data protection regulations such as the European Union's General Data Protection Regulation (GDPR). In addition, the limitations of centralized ML models in terms of storage and processing capabilities have become increasingly apparent (Li et al. 2020a).

To address these concerns, Google introduced federated learning (FL) in 2016 as a distributed learning paradigm that enables training models while preserving the participants' privacy (McMahan et al. 2017). FL updates the global model on the participants' premises with their own data so that only the model updates are transferred to the central server, which aggregates them to create the next-iteration global model (Konečnỳ et al. 2016). This iterative process goes on until the global model converges.

In what follows, FL participants will also be called clients, and the FL server will also be called the model manager. FL, known as "bringing code to the data" rather than "bringing data to the code", addresses data privacy issues and respects the locality and ownership of data (Yang et al. 2019). FL has been successfully implemented in various fields such as finance, healthcare, smart cities, visual object detection, transportation, and next-word/character detection (Long et al. 2020; Xu et al. 2021; Chen et al. 2020; Jiang et al. 2020; Liu et al. 2020; Hard et al. 2018).

Despite its many benefits, FL is not without challenges in terms of privacy and security. Even though the data do not leave the participant's device, model updates computed on those data may still lead to personal information leaks if subjected to privacy attacks (by either honest-but-curious servers or other clients).

Research has been focusing on four main aspects of FL to make it more effective, i.e., the model, the network aspects, the influence of data, and privacy and security:

- Researchers continue to develop new algorithms for FL that can improve the model's performance, in terms of achieving better accuracy.
- Another concern is to solve network-related issues, e.g. reducing communication costs, unreliable connections, and client dropouts. We group the network constraints, bandwidth limitations, and improvements under the network aspects (Yan et al. 2022; Kanagavelu et al. 2022; Zhao et al. 2022).
- FL models are trained on decentralized data, which are often non-independent and identically distributed (non-IID) (Wang et al. 2022). Non-IID refers to the training data being distributed over clients in a heterogeneous fashion. This can make it difficult for the models to generalize well to new data.
- Protecting the privacy and security of client data is a critical consideration in FL. In a federated setting, security can be an issue as clients may modify their updates to poison the model or prevent it from converging. On the other hand, privacy issues come from

the fact that model updates may still leak clients' information, especially in front of curious servers that may conduct inference attacks. Adversaries have been aiming to exploit data points in model training, as recent studies have highlighted issues related to security attacks and privacy leaks (Sun et al. 2021; Aljaafari et al. 2022; Fang et al. 2020; Tolpegin et al. 2020). Moreover, there is a trade-off between performance and privacy. The use of more data for training can improve the performance of the model, but it can also increase the risk of data breaches and unauthorized access to sensitive information. Therefore, studies have aimed to find a balance between utilizing enough data to improve performance and protecting the privacy of the data (Shinde et al. 2021).Natural language processing (NLP) develops algorithms and models to analyze, understand, and generate human language in a way that can be used for various applications, such as language translation, sentiment analysis, and text generation. Text is the natural means of human communication, and textual data commonly convey personal information. Therefore, many NLP applications have rapidly shifted from a centralized setting to the FL setting, where user-generated textual data (such as keyboard typing, web queries, etc.) do not need to be transmitted to a central server. Examples include machine reading comprehension, named entity recognition, next-word prediction, emotion recognition, sentiment analysis Prabhu et al. (2021), and text classification (Ait-Mlouk et al. 2022; Kanani et al. 2022; Zhao et al. 2022; Chhikara et al. 2020; Florea et al. 2021).

The advancement of NLP has been driven by the availability of large-scale text datasets and the development of sophisticated ML models. However, the collection and annotation process of these datasets raises serious privacy concerns. To address these issues, FL presents an appealing alternative by allowing the training of models on decentralized data without the need for data collection or sharing. Whereas FL has been surveyed in previous works, none of them have done it with a focus on NLP tasks.

In this paper, we aim to provide a comprehensive analysis of the current state of the art in FL-based NLP applications, with a focus on model improvement, security and privacy, network aspects enhancement, and handling heterogeneous text datasets. Our goal is to shed light on the strengths and limitations of FL for NLP and present the current open challenges.

Our work brings the following contributions:

1. We are the first to conduct a systematic literature review (SLR) on FL-based NLP applications.
2. We have adopted a multi-dimensional analysis approach to cover the most relevant aspects that influence FL's practicality, i.e., model, network, data, privacy, and security.
3. We present a theoretical analysis, as well as a comparison of the empirical results reported by the considered works on standard datasets.
4. We identify outstanding research challenges.The rest of this paper is structured as follows. Section 2 discusses related surveys on FL. Section 3 presents the search methodology adopted for this study and the research questions we aim to answer. Section 4 provides background on the topics of the survey. Section 5 comprehensively surveys and compares the selected papers from different perspectives, whereas Sect. 6 summarizes the codebase accessibility and other practical details of the surveyed papers. Section 7 discusses the findings of the study and identifies open challenges. Section 8 is a conclusion.

2 Related work

In this section, we discuss the most recent and relevant survey papers on FL. Lim et al. (2020) survey FL in mobile edge networks, highlighting the implementation challenges and reviewing existing solutions with a focus on communication cost, model selection, and resource allocation. Although this comprehensive survey presents the applications of FL for resource optimization in mobile edge networks, it lacks discussion on crucial aspects like security, privacy, and data influence. AbdulRahman et al. (2020) fills the aforementioned gaps and presents a survey that focuses on the shift from centralized to distributed on-site learning. It classifies FL techniques and reviews FL research from three perspectives: model, resource management, and privacy and security. Aledhari et al. (2020) also author a survey on FL examining enabling technologies, applications, and protocols, reviewing the protocols and architectures of FL models, and providing an overview of open challenges. However, a discussion on data influence and the applications of FL in NLP is not given.

Blanco-Justicia et al. (2021) review the security and privacy challenges in FL and survey solutions proposed in the literature up to 2021, but does not deal with NLP. The authors highlight the difficulty of simultaneously achieving both security and privacy protection and provide suggestions for future research in this area. Mothukuri et al. (2021) also survey the security and privacy of FL, and they conclude that security issues are more prevalent than privacy concerns; yet, their study overlooks the coverage of FL models, network aspects, and data influence. Zhu et al. (2021) attempt to fill this gap and discuss FL models and data influence on non-IID data, and they analyze the challenges and influence of non-IID data on FL. However, their study does not cover security and privacy aspects. Shyu et al. (2021) systematically review the advancement of FL in healthcare, and they discuss data influence, security, privacy, data protection challenges, and future research directions. However, their study lacks a discussion on FL models. Liu et al. (2021a) provide the most comprehensive survey to date on FL for NLP: they cover various NLP applications and they identify a few key challenges. However, their work lacks a systematic methodology for paper selection, overlooks the security risks inherent in FL settings, and provides a limited discussion of privacy-related issues.

Soltani et al. (2022) survey FL in mobile networks, specifically focusing on participant selection techniques, their challenges, and future directions. However, their study does not cover FL from the model selection, data influence, security, and privacy perspectives. Extending the aforementioned works, Banabilah et al. (2022) provides a comprehensive review of the fundamentals of FL, including its technologies, privacy concerns, challenges, and future trends. Yet, the authors overlook the data influence and network aspects. Gosselin et al. (2022) survey the privacy and security issues in FL, identifying current issues and their countermeasures. Lastly, Qammar et al. (2022) present a survey that elaborates on the attack surfaces and challenges of FL, providing a taxonomy of attacks and countermeasures concerning privacy, availability, and integrity.

As depicted in Table 1, even though the surveys mentioned above provide valuable insights into FL, they lack a systematic focus on specific applications such as NLP, image processing, or speech recognition, or they do not cover all the relevant perspectives of FL. In this work, we aim to fill the gap by providing an SLR specifically aimed at FL-based NLP applications.

Table 1 Comparison of the coverage of our SLR with related surveys in the same field

Work	Textual data	SLR	Coverage			
			Model	Network aspects	Data influence	Privacy/security
Lim et al. (2020)			✓	✓		
AbdulRahman et al. (2020)			✓	✓		✓
Aledhari et al. (2020)			✓	✓		✓
Blanco-Justicia et al. (2021)						✓
Mothukuri et al. (2021)			✓			✓
Zhu et al. (2021)			✓		✓	
Shyu et al. (2021)		✓			✓	✓
Liu et al. (2021a)	✓		✓	✓	✓	✓
Soltani et al. (2022)				✓		
Banabilah et al. (2022)			✓			✓
Gosselin et al. (2022)			✓			✓
Qammar et al. (2022)		✓	✓			✓
Ours	✓	✓	✓	✓	✓	✓

We uniquely contribute an SLR specific to FL-based NLP. Our work offers a comprehensive and methodical examination of textual data, models, network aspects, data influence, and privacy/security in the field

Table 2 Summary of all search terms, divided into two columns, one for privacy and security, and the other for NLP applications to cover nuanced searches on these topics

Term 1	Term 2
Priva* (private, privacy)	NLP OR natural language pROCESSING
Preserv* (preserve, preservation)	Text analytics/mining
Anonymiz* (anonymize, anonymization)	Sentiment analysis
Secu*(secure, security, securing)	Text classification
Personal	Text segmentation
Sensitive	Text summarization
Perform* (perform, performance)	Text prediction
Comput* (compute, computation)	Text processing
Bottleneck	

3 Search criteria of the SLR

SLRs aim to identify, evaluate, and analyze the entire set of research studies available in a particular research area. It is compulsory that an SLR be carried out using an exhaustive, impartial, and fair search strategy. We approach the analysis of the state of the art by following Kitchenham's preferred reporting items for systematic reviews and meta-analyses (PRISMA) methodology (Moher et al. 2009). PRISMA is a set of guidelines for reporting systematic reviews and meta-analyses, aimed at improving the transparency and completeness of published articles in a field. It provides a checklist of items to be included in a manuscript, covering study design, search methods, data extraction, syntheses, etc.

Table 2 presents the details of the search strings that were used to retrieve articles from Web of Science, Scopus, Google Scholar, and Arxiv. We followed the technique used by Sousa and Kern (2023) and presented two categories of terms. "Term 1" contained privacy

and security-related or performance-related terms, and "Term 2" incorporated terms that exhausted NLP-related tasks.

As a search process for this SLR, articles from archived and published repositories were explored to highlight the most recent trends in FL for NLP. Figure 1 presents the PRISMA diagram for the selection and screening process of the articles.

Articles were screened out at every stage and terms such as "security", "privacy", "metrics", "performance", "NLP", and "evaluation" were considered critical. The stages of the PRISMA diagram were divided into four parts, (a) identification, where overall 805 results were produced from Scopus, Web of Science, Google Scholar, and Arxiv between the years 2018 and 2023; (b) screening with a filter based on title, abstract, and the full text, from which 126 papers were shortlisted, (c) inclusion criteria for eligibility, which resulted in 56 papers being kept, and (d) finally, after several screenings and checks such as language issues, insufficient data, and poor quality of the papers, 36 papers were subjected to analysis.

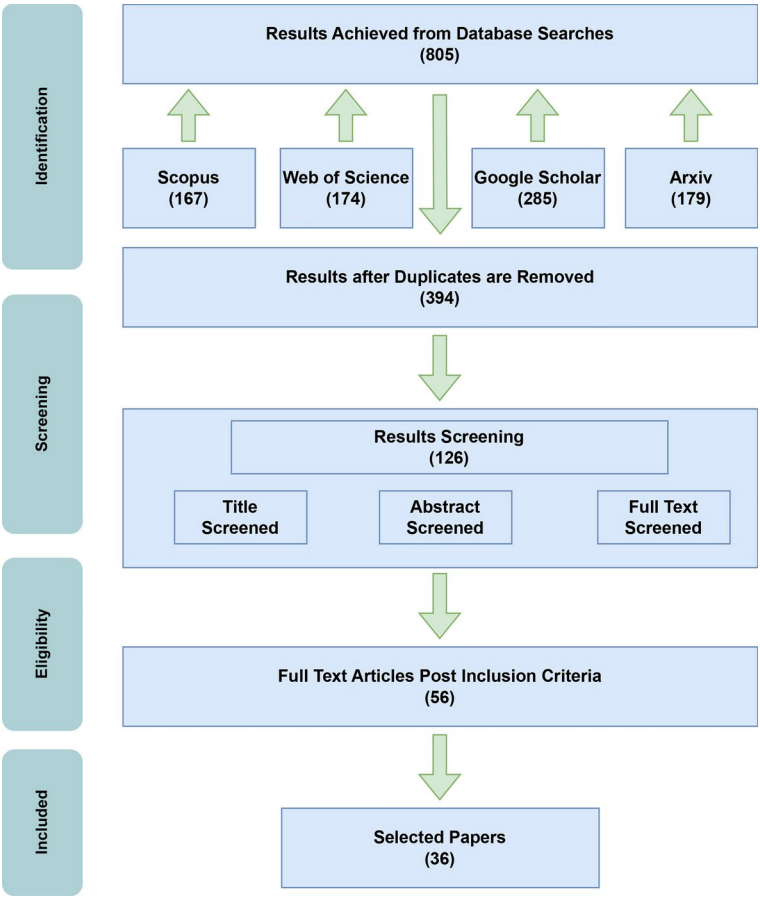

Fig. 1 PRISMA flowchart illustrating the systematic literature search and selection process for identifying relevant papers. Initial database searches yielded 805 results, which were reduced to 394 after removing duplicates. Subsequent screening of titles, abstracts, and full texts, along with the application of inclusion criteria, resulted in the selection of 36 papers for final analysis

3.1 Inclusion criteria

In this section, we discuss the inclusion criteria for systematically selecting research articles.

- Papers published between 2018 and 2023
  - Given the relatively recent introduction of FL, the above range of years should cover relevant works in this area, and hence our survey should be complete in terms of coverage.
- Studies based on FL for NLP
  - We considered only those works that employed FL frameworks in NLP applications.
  - Works should focus on improving any significant aspect of the FL architecture i.e., the trained model, the dataset, the network aspects, or the privacy and security.
- Journals, conferences, and Arxiv articles
  - We wanted to make sure that only scientifically-relevant works were considered. To that end, we selected papers published in peer-reviewed venues. For Arxiv, we selected papers with, at least, 5 citations per year.
- Benchmark datasets
  - To ensure the reproducibility of the reported results, we focused on those papers that used standard and publicly available benchmark datasets in their experiments.

3.2 Research questions

The research questions that our SLR aims to answer are the following.

RQ1: What are the most frequent NLP applications adopted in FL? Which textual datasets have been mostly used?
RQ2: What are the most recent security and privacy attacks on FL for NLP? What are the countermeasures against these attacks?
RQ3: Which network-related factors affect FL for NLP and how does research cope with them?
RQ4: Which data factors influence FL-based NLP? How does recent research tackle them?
RQ5: What are the current open research challenges in FL for NLP?

4 Background

In this section, we briefly discuss the foundations of FL, optimization techniques, network aspects, privacy and security attacks, and evaluation metrics relevant to these dimensions of FL for NLP. These preliminaries are significant for the survey and constitute the basis for the next section.

4.1 Foundations of FL

FL is a distributed ML paradigm that addresses privacy concerns by empowering the participants to train models collaboratively without requiring them to outsource their data to a central server. An FL system has two main actor categories: the clients (data owners) and the FL server (model owner or aggregator). Let $N = \{1, 2, \ldots, n\}$ denote the set of n clients where client $i \in N$ has data D_i, trains an ML model M_i locally, and sends the model parameters to the FL server. The FL server aggregates the received local model parameters as $M = \bigcup_{i \in N} M_i$ and generates a global model M_G as depicted in Fig. 2. The mechanism of action for model training in an FL-enabled system comprises the following steps:

1. *Initialization:* the server sends the initial global model parameters (M_G^0) to all client nodes.
2. *Local training:* at iteration t, each client i trains their new local model M_i^t using their local data D_i and the received M_G^{t-1}. The update of the local model parameters (parameter update from M_G^{t-1} to M_i^t) is then sent to the server.
3. *Global aggregation:* the server updates the global model M_G^{t-1} into M_G^t using the aggregation of the local model updates received from the clients.Steps 2 and 3 are repeated until the global loss function converges or the desired accuracy is achieved.

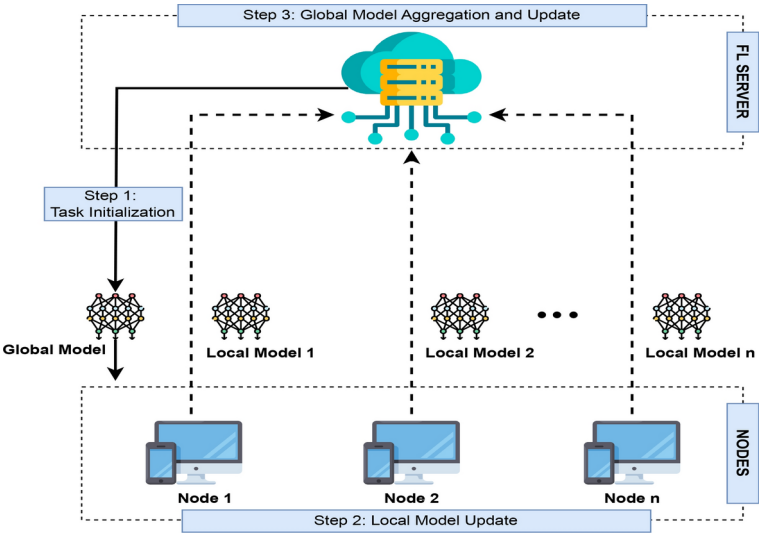

Fig. 2 Workflow of FL, where a global model is distributed to multiple client nodes, enabling them to train the model on their local data. The local models are aggregated to update the global model. This iterative process continues until convergence, empowering collaborative training while protecting local data

This FL process applies to various ML models employing stochastic gradient descent (SGD) for optimization (Lim et al. 2020). Several global model aggregation techniques have been proposed. We describe the most widely used below:

- *Federated averaging* (FedAvg): the FedAvg algorithm calculates a weighted average of client updates based on the proportion of data at each client node (McMahan et al. 2016). It is suitable for IID data, but its convergence is affected by non-IID data.
- *Federated proximal* (FedProx): FedProx adds a regularization term to the local loss function to address FedAvg's slow convergence (Li et al. 2020b). This term minimizes the difference between the global model and the current local model.
- *Federated adaptive* (FedAdam): FedAdam incorporates adaptive learning rates by dynamically adjusting model parameters using gradient magnitudes and convergence rate (Reddi et al. 2020). This approach manages heterogeneous and non-IID data more effectively than FedAvg.

4.2 Network aspects

Since FL models usually have millions of parameters, their deployment in wireless networks usually causes communication and computational bottlenecks due to the limited computational capabilities of the participating client devices (Wang et al. 2019). We briefly discuss network-related constraints of FL settings:

- *Communication* takes place intensively between clients and the server due to the need to broadcast updated models at each iteration, which leads to an enormous cost especially if the model is large. If we talk about pre-trained language models usually employed in NLP, their sizes have increased as they contain billions of learnable parameters. Despite having the upper hand in performance, huge models are unsuitable for practical deployment in federated settings (Wu et al. 2022).
- *Computation* influences the effectiveness of FL systems, especially when deployed on a network with cellular devices. This affects especially the training process, where the heterogeneity of the computational capabilities of the clients may cause delays in updating the global model (Nishio and Yonetani 2019).
- *Dropout* occurs when clients become unresponsive or lag while preparing and sending updates to the server. This may occur for many reasons, such as insufficient communication bandwidth and computation, and variance in data. Dropouts or straggling usually happen in real-world applications of FL when the participating devices are mobile, when clients do not respond during training, or when they are not accessible at certain windows of time (Chen et al. 2020).

4.3 Security and privacy in FL

FL is prone to many types of security attacks, including Byzantine and poisoning attacks. In FL, Byzantine attacks are a type of malicious behavior where participants inject random or arbitrary updates to the model to hamper its convergence, as depicted in Fig. 3. The goal

of these attacks is to cause the model to be inaccurate, even if most participants are honest and provide correct updates (McMahan et al. 2017). On the other hand, poisoning attacks in FL can pose a significant threat to the accuracy and reliability of the learned model (Biggio et al. 2012). These attacks occur when malicious participants inject biased updates into the training process, causing the model to make incorrect predictions, as illustrated in Fig. 4. Such incorrect predictions can lead to biased decisions in real-world applications, which can have serious consequences.

Another security attack on FL is the backdoor attack that can be launched by developing malicious updates. A backdoor is a special part of an input to an ML task (e.g., a specific set of pixels in an image) that triggers a certain ML output selected by the adversary. In the training process, the adversaries possess control over their local datasets, their models' parameters, learning rates, and epochs. Hence, they can influence the global model after aggregation by developing backdoored updates that are incorporated into the learned model after aggregation with genuine updates.

As surveyed by Blanco-Justicia et al. (2021), selecting a large number of epochs, clients, and good clients can help in protection against Byzantine attacks. A statistical comparison of model updates with good or potentially good clients can help detect malicious clients. In order to cope with poisoning attacks, a group of validating clients can be used to determine whether an update from the global model in a particular round is poisoned or not (Andreina et al. 2021). Moreover, Fung et al. (2020) tackled poisoning attacks by evaluating the cosine similarity of clients' previous updates. They term clients as poisonous when the history of their updates is too similar.

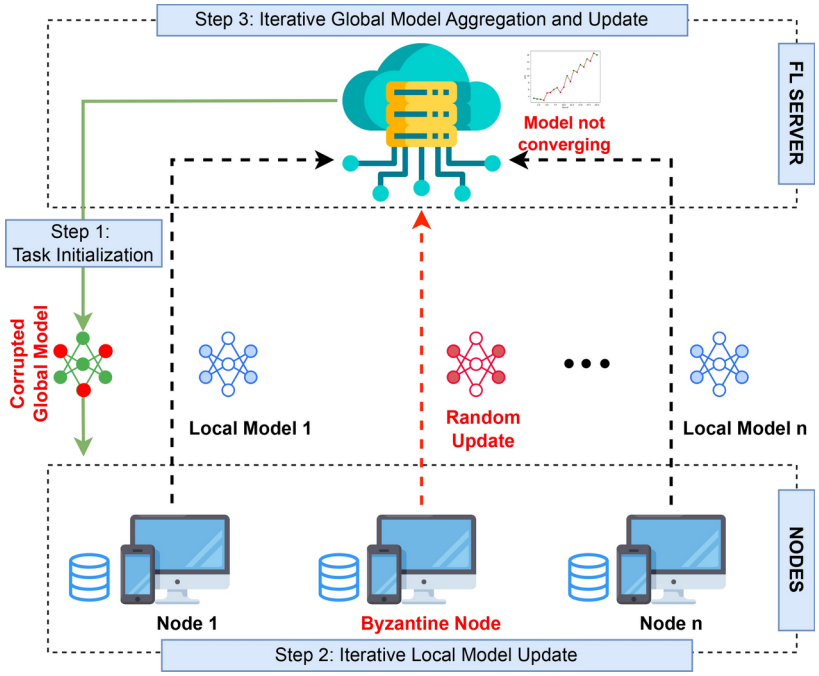

Fig. 3 Byzantine attack, whereby a node introduces random (malicious) model updates to disrupt the convergence of the global model

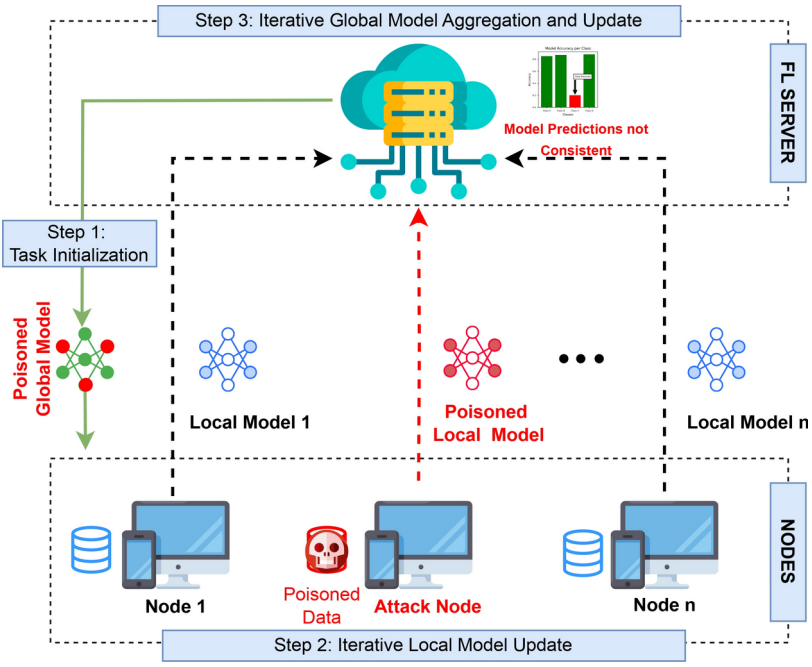

Fig. 4 Poisoning attack, whereby a node injects targeted poisoned data to manipulate the global model's predictions (e.g. for a certain class)

Backdoor attacks in FL can be mitigated by secure aggregation algorithms using cryptographic techniques to ensure the privacy and security of the updates during the aggregation process and to detect any malicious updates that may contain backdoors (Bonawitz et al. 2017). Adversarial training is another way of dealing with backdooring. It involves training the model to resist attacks from malicious participants, by adding adversarial examples to the training data and adjusting the model parameters accordingly (Wang et al. 2019).

Recently, Jebreel et al. (2023) have proposed a defense against backdoors that distinguishes backdoored updates from benign updates by looking at their effect on specific layers of the learned model.

The decentralized nature of FL also raises several privacy concerns. One of the privacy attacks in FL is the model inference attack. An adversary can use the trained model to infer sensitive information about a client's data by examining the model parameters (Fredrikson et al. 2015). Model inversion attacks try to recover the client's private dataset used to generate a certain update.

Attribute inference is another privacy attack in FL that can use the model parameters trained in FL to infer sensitive information about the clients or about the subjects the clients' local data refer to, such as demographic information, health status, and other personal attributes. The adversary can use the correlations between the model parameters and the sensitive information to infer the sensitive information.

Finally, membership inference attacks are used to determine whether a specific data sample was used to train the model in FL. This type of attack can reveal the presence of a

specific subject's record in a client's local data, which can result in privacy breaches (Shokri et al. 2017).

Several techniques have been proposed to counter these attacks. For example, model inversion attacks are mitigated by adding random noise to the model parameters during the local model update. To avoid inference attacks, differential privacy (DP) is commonly used to enhance privacy in FL. The use of DP results in distorted client updates so that the absence or presence of a specific record in a client's local data does not notably impact the update. Henceforth, no straightforward inferences can be made by the FL server or adversaries on the clients' data through individual updates (Blanco-Justicia et al. 2021). However, the accuracy of the model is significantly degraded by the added distortion (Domingo-Ferrer et al. 2021). One way to mitigate membership inference attacks is to use random sampling and aggregation techniques to prevent an adversary from determining which data samples were used to train the model in FL (Shokri et al. 2017).

4.4 Evaluation metrics

We briefly describe the evaluation metrics referenced throughout the paper to assess the performance, privacy, and security capabilities of FL systems for NLP tasks.

1. *Model-related metrics:*

   - *Accuracy (Acc.), average accuracy (Avg. Acc), and top-k accuracy (Top-k Acc.):* these metrics measure the overall correctness of the model, the mean accuracy achieved across multiple runs, and the model's ability to rank the correct class within the top-*k* predicted classes (e.g., $k = 1$ or $k = 50$), respectively (Murphy 2012).
   - *Prediction recall:* this quantifies the model's accuracy in predicting the next word, emoji, etc., and measures the model's ability to make correct predictions given distributed data from multiple clients (Hard et al. 2018).
   - *F1-score:* this is the harmonic mean of precision (the correct identification of positive instances) and recall (the ability to find all positive instances). F1-Score provides a balanced view of a model's performance by considering both precision and recall (Murphy 2012).
   - *Mathews correlation coefficient (MCC):* MCC is used to evaluate the quality of binary classification predictions. It considers true/false positives and negatives for a more robust evaluation, especially in imbalanced datasets (Chicco and Jurman 2020). Higher MCC values indicate better classification performance.
   - *Perplexity:* perplexity measures the difficulty of predicting the next word in a sequence, where a lower value indicates better model performance. Average perplexity (Avg. Perp.) refers to the mean perplexity calculated across the test data (Jelinek et al. 1977).

2. *Network-related metrics:*

   - *Communication rounds (Comm. rounds):* this term refers to the number of communication rounds required between the FL server and the participating nodes for the model to converge, where fewer rounds mean better efficiency (Mills et al. 2023).

3. *Security and privacy-related metrics:*

- *Attack success rate (ASR):* ASR is the proportion of targeted examples (with the source label) successfully misclassified into an attacker's desired label (Jebreel et al. 2024). For example in sentiment analysis, a "positive" review is changed to a "negative" one. A higher ASR indicates a more effective attack, while a lower ASR indicates a more effective defense.
- *Recovery rate:* recovery rate is the maximum percentage of tokens in the ground truth recovered by the attack algorithm (Deng et al. 2021).
- *Leakage:* Maheshwari et al. (2022) define leakage as the accuracy of a classifier in predicting sensitive attributes from their encoded representations. The lower this accuracy percentage, the stronger the protection, that is, the smaller the ability to infer sensitive information.
- *Success rate:* according to this metric, an attack is successful if the mean squared error between the recovered samples (from gradients) and the original input is ≤ 0.001 (Huang et al. 2020).

5 FL for NLP

In this section, we present a comprehensive survey of 36 selected papers on FL techniques for NLP tasks. We structure these techniques along four dimensions crucial to the successful deployment of FL for NLP: NLP applications, security and privacy, network aspects, and the influence of data.

We explore various NLP tasks where FL has demonstrated improved model performance and convergence speed. We also review how FL techniques have addressed challenges related to privacy and security, protecting against information extraction, and ensuring system integrity. In examining network aspects, we investigate how communication, computation costs, and other relevant factors associated with FL are enhanced. Finally, we explore how the nature of data, such as class imbalance and non-IID distributions, affects FL settings. The analysis conducted here answers research questions RQ1 to RQ4, listed in Sect. 3.2. Figure 5 presents the taxonomy of FL methods for NLP that serves as the framework for our SLR.

5.1 NLP applications

FL has been used in several NLP tasks. Many of these tasks often involve NLP models with many parameters, which makes them computationally intensive to train. FL enables clients to perform the training locally, thereby reducing the computational burden on the central server and making the training process more scalable. Most of the works surveyed here can be categorized into one of the following general tasks: prediction, classification, and sentiment analysis.

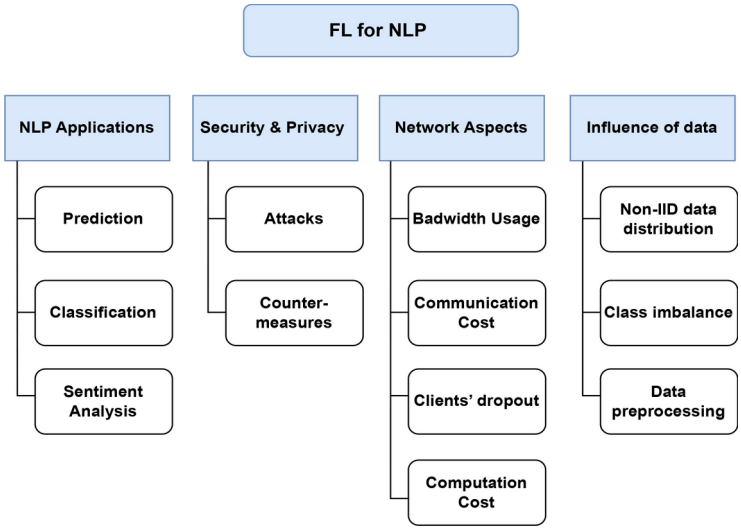

Fig. 5 Taxonomy of the surveyed research, encompassing four key dimensions, i.e. NLP applications where FL is used, security and privacy attacks and their countermeasures in FL for NLP, network-related factors that impact FL performance, and lastly, data attributes that influence the effectiveness of FL for NLP tasks

5.1.1 Prediction

Text prediction is a task focused on analyzing fine-grained client inputs. For example, inputs on a keyboard are especially sensitive to privacy issues (in fact, they act like keyloggers, which constitute a serious privacy threat). Hence, the application of FL to such NLP tasks is common and useful. We identified five prediction tasks such as the next word or character, emoji, query, and rating predictions, in the surveyed papers for which FL is practically used.

Yang et al. (2018) at Google improved the quality of search suggestions in virtual keyboards, using FedAvg with LSTM (long short-term memory) (Hochreiter and Schmidhuber 1997) in a global-scale setting. They collected real user interaction data from the Gboard (Google Keyboard, a virtual keyboard for cell phones). Specifically, they used data from Gboard users who opted to share anonymized snippets of text typed in selected apps periodically. They stripped the personally identifiable information from these logs and used a subset of logs for training. The goal was to suggest a query to the user and then observe the click-through rate (CTR), which was reported to be 51.49%. We refer to CTR as accuracy in Table 3. Their major limitation is that they only used English sentences.

By leveraging anonymized user interactions within Gboard in Yang et al. (2018), Hard et al. (2018) used a variant of the LSTM recurrent neural network (RNN, Rumelhart et al. (1986b)) called CIFG (coupled input and forget gate) for next-word prediction. It was trained with FL and server-based SGD. The CIFG model achieved prediction recalls of 27% and 15.8% for server-hosted and client-owned data caches, respectively, and 13.75% for live user traffic. While the model performed well, its performance degraded by 50% compared to the server-hosted logs.

Ramaswamy et al. (2019) built upon previous practical applications of FL by Hard et al. (2018) and used a word-level RNN to predict emoji from the text on a mobile keyboard.

Table 3 Summary of FL techniques for prediction tasks

Dataset	References	Focus	Evaluation	Side effects/limitations
Logs	Yang et al. (2018)	Model improvement	Accuracy = 51.49%	Non-English text not considered
	Hard et al. (2018)	Model improvement	Prediction Recall = 13.75%	Lower performance than server-based
	Ramaswamy et al. (2019)	Model improvement	Top-1 Acc. = 25.6%	Non-English text not considered
Stack overflow	Stremmel and Singh (2021)	Model improvement	Accuracy = 25.69%	Only feasible for similar datasets
Amazon	Liu et al. (2021b)	Model improvement	F1-Score = 0.0138	Data & objective heterogeneity impacted it
Shakespeare	Li et al. (2021)	Model improvement	Accuracy = 58%	Led to privacy loss
	Dudziak et al. (2022)	Model improvement	Avg. Perp. = 3.4	Led to privacy loss
	Mills et al. (2023)	Accelerate convergence speed	Comm. rounds = 3000	Led to increased computational cost

Most studies focus on improving model performance or system efficiency. They achieve several notable improvements, but several challenges remain: limited consideration of non-English text, privacy concerns, high computational cost, or issues with data heterogeneity in diverse datasets

They pre-trained the model using transfer learning and proposed mechanisms for triggering and tuning the diversity of emoji candidates. They used the FedAvg algorithm with CIFG and achieved an accuracy of 25.6% for the best federated model and 23.9% for the best server-trained model. The dataset comprised approximately 370 million snippets, 11 million of which contained emoji. The logs were filtered to include only highly confident English sentences by a language detection model. One potential limitation of their work is the lack of data from non-English languages, which may affect the generalizability of their model to other languages.

Stremmel and Singh (2021) built on the existing body of FL experiments, with a focus on enhancing accuracy and reducing the required number of training rounds for federated text models. For the next-word prediction task using Stack Overflow by Oktay et al. (2010), they used federated fine-tuning with pre-trained text models and generative pre-trained transformer 2 [GPT2, Radford et al. (2019)] word embeddings. As text data usually have a large frequency gap between the most common and least common words, they limited the vocabulary size to exclude rare words in their experiments. They found that pre-trained word embeddings generally outperformed random embeddings after training for 1,500 rounds,

with evaluation on 10,000 validation samples per training round. They performed a final evaluation by averaging the last 100 rounds of validation accuracy without special tokens. Their approach achieved a better accuracy of 25.69% compared to the random technique's 24.85%. However, a limitation of their work is that the same model could not perform well on the Shakespeare dataset of McMahan et al. (2017). They mentioned that the pre-training data for Shakespeare was not similar to the Stack Overflow data.

Liu et al. (2021b) investigated rating prediction using the Amazon rating data by Ni et al. (2019). The dataset included user-item-rating triplets for 29 different domains. To examine the impact of the number of domains on the model performance, they selected the four most populated domains (Books, Electronics, Home, and Clothing). They proposed a technique to learn and sustain decentralized user encodings on the user's personal space. They showed that learning on smartphones or laptops could resolve the privacy concerns of direct information sharing in cross-domain recommendation techniques. Their technique performed better regarding recommendation and prediction for cold start users. The model performance was influenced when heterogeneity in terms of data and objectives was introduced.

The works that follow are dedicated to exploring specific aspects of next-character prediction for mobile devices using the Shakespeare dataset. Li et al. (2021) focused on optimizing communication and computation costs, energy consumption, memory footprint, and inference accuracy for mobile devices. Their FedMask approach did not explicitly leverage pre-trained embeddings but aligned with the practical considerations raised by Stremmel and Singh (2021).

On the other hand, Dudziak et al. (2022) proposed a novel aggregation method in FL that enables weight sharing among clients. This method significantly improved how well the model predicts the next character compared to the state of the art (12.58% perplexity improvement). However, they acknowledged potential privacy concerns that require further investigation. Lastly, Mills et al. (2023) tackled the challenge of accelerating the speed of convergence within this context. By leveraging LEAF (a benchmarking framework for learning in federated settings Caldas et al. (2018)) for data pre-processing, they were able to efficiently train models. However, their approach, while effective, increased computational cost, which makes it less suitable for mobile devices with limited resources.

In conclusion, these studies present various techniques for improving FL models for text prediction tasks. These techniques aim to speed up convergence or enhance model accuracy. However, the improvements may come at the expense of privacy and further research is needed to ensure a balance between performance and privacy in FL for text prediction. Table 3 provides a summary of the evaluation figures reported by all the studies on prediction tasks in an FL setting. We have listed the datasets that are used in the study along with the focus of the research, and the side effects (in case of improvement) or limitations (if any).

5.1.2 Classification

We have identified four types of classification tasks in the surveyed papers, i.e., short message service (SMS) spam classification, document classification, sentence-level classification, and topic classification. SMS spam classification is used to determine whether a given message is spam or not. Document classification works on a given text (news article, tweet, etc.) and classifies it into a predefined set of categories such as political, sports, entertain-

ment, etc. Sentence-level classification is used to classify a given sentence as grammatically acceptable or not. Last, topic classification discovers latent topics in documents based on word frequency, order, distance, and meaning.

Tran et al. (2021) utilized the UCI Spam Collection dataset (Almeida et al. 2011), an imbalanced dataset of SMS-tagged messages. They proposed and evaluated a secure decentralized training framework (SDTF) for SMS spam classification. They compared the performance of their SDTF against a centralized LSTM model over 100 communication rounds, with the SDTF achieving 92% accuracy and the LSTM reaching 98.73%. Extending the investigation of FL for this task, Florea et al. (2021) also leveraged the SMS Spam Collection dataset. They focused on its shorter texts to benchmark the performance of SL against a centralized setting. Their FL model demonstrated high accuracy but encountered scalability challenges. It required 12 times longer training time than the centralized model due to the use of only two workers (Bob and Alice) and it resulted in communication overhead.

Lin et al. (2022) conducted several experiments using consistent settings (communication rounds and data partitions) to benchmark. They employed text classification models to classify news articles as political, sports, entertainment, etc. They found that FedOPT outperformed FedProx and FedAvg on NLP tasks, including classification. They used 20 News (Lang 1995) and other datasets for document classification with annotations for 20 labels. They showcased their use of FedNLP with this dataset as it had a larger output space, which was an essential factor for label-distribution shift scenarios. Cai et al. (2023) found that FedNLP's large model size and high computation and network costs resulted in slow performance. To address this issue, they proposed AutoFedNLP, which solved the problem of adapter depth and width configuration. The proposed framework progressively enhanced adapter configurations during the training process. The researchers suggested learning fewer models for shallow knowledge, and then incrementally acquiring deep knowledge with larger and deeper adapters. A major limitation of both these works is that they did not consider the impact on privacy and security.

For sentence-level classification, Li et al. (2022b) presented and evaluated FedSplitBERT using the corpus of linguistic acceptability [CoLA, Warstadt et al. (2019)]. Their methods could fine-tune bidirectional encoder representations from transformers [BERT, Devlin et al. (2018)] over multiple clients with heterogeneous data, resulting in better performance than baseline methods such as FedAvg, FedProx, and FedAdam. However, they did not discuss the impact of their technique on communication and privacy.

Lastly, for topic classification, Si et al. (2022) presented a collaborative approach for training a non-negative matrix factorization (NMF) topic model across several clients using FedNMF and mutual information maximization. They evaluated their model by utilizing latent document representations for external tasks, such as text classification via support vector machines (Hearst et al. 1998), and by creating various FL settings. Their approach demonstrated effectiveness in collaborative topic modeling, but they did not address the privacy implications of their technique.

Table 4 summarizes studies that have worked on the classification task in a federated setting. We group the studies based on the dataset they used. We also present the focus of the research and its evaluation metrics. Finally, we mention the side effects or shortcomings (if the study tries to improve the model) or the limitations of each paper (if any).

FL for classification tasks has primarily focused on improving model performance by enhancing accuracy and convergence speed. However, such improvements may come at the

expense of security, privacy, or communication costs. It can be observed that most of the techniques discussed here focus on the mitigation of the problems caused by non-IID data in NLP tasks, which is a significant concern in FL. Despite efforts to enhance performance, FL for classification still lags behind centralized systems in terms of performance.

5.1.3 Sentiment analysis

In the selected papers, we analyzed studies on three types of sentiment analysis tasks: binary sentiment analysis (positive, negative), ternary sentiment analysis (positive, negative, neutral), or sentiment analysis with more categories.

For binary sentiment analysis, Fu et al. (2021) experimented on four datasets, including the IMDB (internet movie database) dataset by Maas et al. (2011). They employed 20 clients and generated two clusters for the IMDB dataset. They claimed that their model achieved better accuracy and preserved privacy. However, their framework was not analyzed under any threats.

For ternary sentiment analysis, Duan et al. (2021) and Mills et al. (2023) investigated Twitter posts using the Sentiment 140 dataset by Go et al. (2009). Duan et al. (2021) proposed FedGroup, a clustered FL framework that grouped the clients' training based on their optimization similarity. For the scalability and practicality of the framework, they implemented a newcomer device cold-start mechanism. They were able to achieve an accuracy of 76%. Despite reporting good results, neither study explicitly addressed the privacy and security implications of their approaches. In contrast, Mills et al. (2023) focused on accelerating the convergence speed by employing a globally biased optimizer. They discovered that the convergence speed is inversely proportional to the number of parameters in relatively

Table 4 FL techniques for text classification tasks

Dataset	References	Focus	Evaluation	Side effects/ limitations
SMS Spam	Florea et al. (2021)	Comparison	Acc. FL = 96%, ML = 98%	Privacy not considered
	Tran et al. (2021)	Model improvement	Acc. = 92%	Acc. lower than centralized setting (98.73%)
20 News	Lin et al. (2022)	Model improvement	Acc. = 53.49%	Privacy not considered
	Cai et al. (2023)	Convergence speed	Desired Acc. = 99% time taken= 1.3 hrs	Privacy not considered
CoLa	Li et al. (2022b)	Model improvement	MCC = 51.17%	Comm. cost & privacy not discussed
Stack Overflow	Si et al. (2022)	Model improvement	Acc. = 83% F1 = 82%	Impact on privacy was not considered

Most research aimed to improve model accuracy, convergence speed, and communication costs. A common limitation of these works is that they fall short of simultaneously improving the model and addressing privacy and communications challenges

straightforward learning tasks. Furthermore, adaptive optimization increases the computational cost and may not be suitable for mobile devices.

Qin et al. (2021) proposed a novel approach for aspect-based sentiment analysis (ABSA) using FL. Their model incorporated topic memory with BERT to leverage data from multiple sources, which addressed the challenge of diverse sentence structures in ABSA tasks. Building on this foundation, Maheshwari et al. (2022) further explored FL for sentiment analysis, focusing on privacy, fairness, and accuracy. They proposed FEDERATE, a technique that used a DP encoder and adversarial learning to acquire fair, accurate models with representations for identification on four different datasets, including the Twitter Sentiment dataset by Blodgett et al. (2016). Both of these studies reported promising results, but they did not explicitly address the impact of their techniques on communication costs or potential security vulnerabilities.

Lastly, Bhardwaj et al. (2022) curated the Amazon fine food reviews dataset having five-class sentiment classification, with each review rated on a scale of 1 (strongly negative) to 5 (strongly positive). The system considered the potential influence of race-specific words on sentiment prediction, for example, the word "White" often appeared in neutral reviews while "Latino" was associated with positive sentiment. They introduced knowledge distillation using optimal transport (KNOT), an approach for sentiment classification that uses a semantic distance metric to measure the closeness between actual and predicted labels, thus training a global model with minimal transport cost. Their results revealed performance improvements in the global model's semantic distance performance compared to other baselines, achieving results on par with entropy-based distillation. However, privacy was considered out of the scope of this study.

Table 5 summarizes the reviewed works on sentiment analysis in FL. They are grouped by the dataset used and details such as research focus and evaluation metrics are provided. Side effects of solutions and remedies for attacks are also mentioned.

The main focus of research on FL for sentiment analysis tasks is to improve models in terms of accelerating convergence speed and achieving better accuracy. The reviewed studies have reported improvements in accuracy and convergence speed. However, they do not report the impact of their techniques on privacy, security, and computational costs.

5.2 Security and privacy in FL for NLP

NLP is becoming increasingly pervasive in our daily lives, and it also brings new risks and challenges. Attackers are constantly searching for vulnerabilities in NLP systems they can exploit, which makes it crucial for practitioners to implement effective countermeasures. As discussed earlier, FL emerged as a promising approach for training NLP models using decentralized data sources. However, FL is also vulnerable to attacks that can compromise privacy and security. In the following subsections, we explore some of the most common attacks on FL for NLP, and we discuss the countermeasures that can be employed to protect against them.

5.2.1 Attacks on FL

Privacy attacks on FL for textual data aim to extract sensitive information from the text data of individual participants without their consent or knowledge. On the other hand, *security*

Table 5 FL approaches for sentiment analysis across diverse datasets

Dataset	References	Focus	Evaluation	Side effects/ limitations
IMDB	Fu et al. (2021)	Model improvement	Acc. = 83.2%	Privacy & communication costs were not considered
Sentiment 140	Duan et al. (2021)	Model improvement	Acc. 76.1% (72% for cold start)	Privacy was not considered
	Mills et al. (2023)	Accelerate convergence	Comm. rounds = 1000	Higher computational costs
Twitter	Qin et al. (2021)	Model improvement	Acc & F1 = 74% & 73%	Privacy was not considered
	Maheshwari et al. (2022)	Performance	Acc. = 74%	Security & communication costs were not considered
Amazon	Bhardwaj et al. (2022)	Model improvement	Acc. = 59%	Privacy was not considered

The main aim of these works was to enhance model performance and convergence. However, the impact on privacy and communication and computation costs was mostly overlooked

attacks in FL for textual data aim to compromise the integrity or availability of the FL system itself. Here we discuss different attacks on the privacy and security of federated settings for NLP.

The number of attacks on FL for text is relatively small, but the literature contains studies that target FL language model updates. Early work by Yuan et al. (2021) proposed the first record-level attack on a federated setting to disclose clients' identities and extract private records. They investigated record exposures and monitored patterns in them. The study presented two correlation attacks, i.e., eavesdropping and watermarking attacks. They exploited the fact that adversaries can use eavesdropping attacks to access the communication channel between clients and servers in FL, allowing them to retrieve private data. They combined the eavesdropping attack with a watermarking technique to explore record-level leakage in FL without reaching the client's model. The attacker assumed access to the weights of the global model during each aggregation and knowledge of the victim client selection, which was achieved by monitoring communication around the client. By identifying the exposed training data of the selected client, the adversary was able to derive the client's private record using the highest correlation coefficient. If victim selection was unavailable, a watermarking attack could be used to generate a correlation. By injecting a watermark into the victim client's dataset, the attacker could compare the correlation of changes in exposure rates between the watermark and potential records to extract records of interest. However, the attack might deteriorate when bigger datasets are used.

Similarly, Deng et al. (2021) tried to recover the local training data from transformer-based language models. They presented a novel gradient attack on transformer-based language models, TAG, that formulated the gradient attack on transformer-based learning

models, including several BERT models, e.g., BERT large, BERT base, and BERT tiny. During local training, adversaries cannot access private training data directly but can obtain gradients and the global model at any time. Attacking NLP applications was more challenging due to the larger token space, the need for exact matches for sensitive information, and errors in retrieved token IDs leading to irrelevant strings. The attack could happen at any training stage, and the two most common weight initialization methods, random initialization for non-pre-trained models and specific learned values for pre-trained models were considered. The authors claimed to have obtained a cosine similarity of 0.93 and a recovery rate of 89% in token embeddings from private data, which highlights the importance of considering privacy in FL systems. They did not evaluate their model in the presence of defenses.

If we extend the focus on gradient attacks, Gupta et al. (2022) also presented a gradient attack called federated inversion attack for language models or FILM. The proposed attack is used to extract text from large batches. They targeted gradients for word identification and reconstructed sentences based on a reordering strategy. They claimed that their attack worked for a batch size as large as 128 sentences, but the ablation study did not go beyond 16. Rather than matching gradients, they first identified a set of words and reconstructed sentences using a beam search and a prior-based reordering strategy. FILM leveraged prior knowledge in pre-trained language models or memorization during training. Despite its simplicity, FILM worked well with large-scale datasets, extracting multiple sentences with high fidelity and successfully recovering multiple sentences if applied iteratively. The authors did not test their attack in the presence of defenses, even if freezing word embeddings during training can prevent reconstruction.

In contrast to these approaches, Fowl et al. (2023) depicted through their research that FL on text is more vulnerable than previously anticipated because the usual focus is on "honest-but-curious" threat models where the users receive benign parameters. They proposed a new attack that discloses private user texts using malicious parameter vectors. They argued that their proposed threat model was just as realistic as the "honest-but-curious" model since a curious server could send a single malicious parameter vector to users to retrieve private data. The authors demonstrated that privacy inference is a major concern in FL settings, especially with their attack called Decepticons. The attack focused on untrusted server updates and was able to retrieve a large amount of private information by reprogramming transformer parameters and using statistical procedures. The authors discovered that the threat posed by their attack was much greater than the current state of the art in "honest-but-curious" models. They claimed to be able to recover 90% of the tokens for GPT2 in a single sequence, and for multi-sequence the accuracy of the recovery was 50%. However, DP can be used as a defense mechanism against the proposed attack.

On the other hand, backdooring has been used as an effective security attack in FL for NLP settings. Yoo and Kwak (2022) exploited the vulnerability of FL models to such attacks by poisoning them with rare word embeddings and utilizing gradient ensembles (GE). They showed that only 1% of adversary clients can divert the output of the global model in a text classification task. The application of GE increases both the final backdoor performance and its persistence during training. Clean accuracies of poisoned and non-poisoned runs do not differ significantly, making detection through validation difficult. Optimal bounds that preserve clean performance were chosen and tested on a range of values, as was done for DP to determine standard deviation. Coord-median and multi-Krum are more computationally expensive defense techniques that can prevent model poisoning at a realis-

tic adversary-client ratio. However, when tested to defend against the proposed technique, coord-median incurred a longer aggregation time, while multi-Krum experienced decreased clean performance.

By further exploiting the threat of backdoor attacks in FL, Zhang et al. (2022) introduced PipAttack. Their strategy is based on backdooring federated recommender systems to exploit the popularity bias in data-driven recommenders for targeted item promotion. This is achieved by designing an attack model that makes the target item appear as a popular item in the embedding space. The attack is carried out by uploading crafted gradients via a few malicious users during the model update. The attack accuracy of the proposed technique is 100% after 40 epochs at a mere infection of 5%. Evaluations show that the attack significantly boosts the exposure rate of the target item without affecting the accuracy of the poisoned recommender. Existing defenses are not effective against this attack, which highlights the need for new defenses against local model poisoning attacks on federated recommender systems. However, the presence of explicit promotion or density of the data might influence the performance of the attack.

A summary of the privacy and security attacks and their foundational techniques is provided in Table 6.

5.2.2 Countermeasures

Privacy countermeasures for FL refer to the measures taken to protect the privacy of local user data in FL systems. In an FL system, user data are decentralized, and users may have different levels of privacy concerns. On the other hand, security countermeasures for FL refer to the measures taken to protect the FL system from various security threats such as unauthorized access, data breaches, and malicious attacks. FL systems typically involve

Table 6 Privacy and security attacks in FL for NLP

Various attacks, their evaluation metrics, and associated limitations or defense mechanisms are presented. Privacy attacks were mainly gradient-based, whereas security attacks exploited backdooring

Attack	References	Technique	Evaluation	Side effects/ limitations
Privacy	Yuan et al. (2021)	Eavesdropping & Watermarking	Top-50 Acc. = 85%	Big datasets
	Deng et al. (2021)	Gradient-based	Recovery rate = 89%	Not tested with defenses
	Gupta et al. (2022)	Gradient-based	Recall = 34% (100 iterations)	Freezing word embeddings during training can prevent reconstruction
	Fowl et al. (2023)	Honest but curious server	Recovery rate, single & Multi: 90 and 50%	DP can be used as defense
Security	Yoo and Kwak (2022)	Backdoor	Avg. Acc. = 92.57%	Multi-Krum Coord and median as defenses (but they are expensive)
	Zhang et al. (2022)	Backdoor	Acc. = 100%	Explicit promotion & denser data

multiple parties sharing data and models, which increases the risk of security threats. To counter these threats, FL systems may implement numerous countermeasures that seek to keep data and models secure throughout the FL process.

In this section, we survey contributions to defending against privacy and security attacks on FL for NLP.

Huang et al. (2020) addressed the challenge of mitigating privacy issues in a federated environment without compromising the training speed or accuracy of the system. They proposed TextHide, which aims to address privacy risks during the training process. TextHide needs all the participating devices to add an encryption step to protect against eavesdroppers who intend to recover private texts. It is designed to be integrated with popular pre-trained language models like BERT, which transforms textual input into output vectors. It utilizes the output representations generated by the pre-trained encoder to train a new shallow model (e.g., logistic regression, Morgan and Teachman (1988)) for any supervised single-sentence or sentence-pair task. During training, the pre-trained encoder was fine-tuned along with the shallow model. The results revealed that the encrypted representations generated by TextHide were secure, meaning there was no efficient way to recover the original text from the security framework, as it made gradient matching harder. However, if direct training on the encrypted text was enabled, the results in terms of performance could also be improved. The authors evaluated their framework on 50 independent gradient attacks and reported that the success rate was reduced to 8% from 82%.

To overcome the limitations of TextHide, which incurred a substantial computational cost, Tran et al. (2021) proposed the above-mentioned SDTF method. This method did not require a trusted server, and at the same time ensured the privacy of local data. It used an efficient secure sum protocol (ESSP) to enable a huge number of clients to calculate the sum of their private inputs. ESSP enabled secure training and sharing of the local model updates for aggregation. It allowed a large group of parties to calculate a sum of private inputs securely. This protocol was, reportedly, the first of its kind that was able to handle both integer and floating-point numbers without requiring any data conversion. The framework used randomization techniques in combination with ESSP to protect the local models from honest-but-curious parties. The paper presented theoretical proofs for evaluating its model. A major limitation is that it was not tested against privacy attacks.

Maheshwari et al. (2022) also focused on privacy in FL. They used DP as a defense mechanism to protect the privacy of sensitive information in the model. They proposed a generic private encoder construction consisting of two main components. The first component is an encoder that maps the text input to a D-dimensional vector space. It can either be a pre-trained language model with some trainable layers or trained from scratch. The second component is a randomized mapping that transforms the encoded input into a differentially private representation. Through their experiments, they demonstrated that their method had the capability of inducing both private representations and fair models simultaneously. They reported an average leakage of 61.74%. A potential limitation of their work might be fairness, as they did not provide any specific fairness analysis.

Regarding the security of FL, Wang et al. (2021) presented a novel technique called SEFL (secure and efficient FL). The SEFL framework is designed based on the two non-colluding (untrusted) server settings, where an aggregation server aggregates the encrypted local model updates, and another server manages the cryptographic primitives (i.e., the decryption key). This prevents attacks on the model during the aggregation process. The

use of encryption and cryptography helps ensure that the local model updates are protected from interception or modification during transmission, and the non-colluding servers add an extra layer of security to prevent malicious actors from compromising the system. However, the authors did not evaluate their framework by launching potential attacks at it. Instead, they measured its accuracy. They reported a comparison of the perplexities of the pruned and unpruned versions for transformer-based and LSTM-based models. The improvement in perplexities for both models was reported as 19.3% and 12.8%, respectively.

More recently, Jebreel et al. (2024) presented fragmented FL or FFL. The framework improves security by introducing a reputation-based defense mechanism that utilizes fragment quality. This approach helps filter out malicious updates that may compromise the availability and integrity of the model. The reputation-based defense mechanism works by evaluating the quality of each fragment based on the client's reputation. Clients with a good reputation are more likely to have high-quality fragments, whereas clients with a poor reputation are more likely to have low-quality fragments. Overall, the FFL framework's reputation-based defense mechanism helps ensure that the global model remains secure and protected against malicious attacks. By utilizing fragment quality and client reputation, the FFL framework can prevent malicious updates from compromising the security of the model. The authors evaluated their technique against the label-flipping attack and their ASR and test error (the error resulting from the loss functions used in training) were among the best at 13.36% and 0.544, respectively. There is room for improvement as far as the runtime is concerned as there was an overhead in the runtime of FFL.

As summarized in Table 7, cryptographic techniques have been widely employed to safeguard the security and privacy of FL systems dealing with NLP tasks.

Conciliating security and privacy issues in FL for NLP poses challenges due to the data being non-IID and sensitive. For instance, filtering out apparently bad or outlying updates conflicts with legitimately outlying data, especially in non-IID settings. Overall, the challenge in conciliating security and privacy issues in FL for NLP lies in ensuring the security and privacy of the data while maintaining the model's performance and generalizability. Besides the discussed research, Augenstein et al. (2020) proposed using synthetic data generated by differentially private federated generative models to overcome the limitation of data inspection in private, decentralized data settings. The authors aimed to improve the performance, privacy, and security of FL for NLP by generating synthetic data that were representative of private data without compromising the privacy of individuals. FewFedWeight by Dong et al. (2022) also enhanced the privacy and security of data in FL by generating synthetic data and facilitating cross-task knowledge sharing without data exchange. This proposal ensures that sensitive information remains protected and secure throughout the learning process. Moreover, vertical FL was proposed to solve the data heterogeneity problem, where deep models such as SplitNN were successfully applied. However, SplitNN leaks information, as transformed data are directly sent to active parties. Since a correlation between the transformed and original data is found, inference attacks can recover the original data. To mitigate local attack timing, Zawad et al. (2021) proposed an active defense mechanism by maintaining a global IID dataset to minimize overfitting and improve model generalizability. Lastly, Cai et al. (2023) presented a secure forward aggregation (SFA) to mitigate the local attack timing further. SFA used a different way of aggregation for transforming data and a removable mask for data protection, claiming to achieve high model

Table 7 Methods to counter privacy & security attacks on FL for NLP

Counteract	References	Technique	Evaluation	Side Effects / Limitations
Privacy	Huang et al. (2020)	Cryptography	Success rate = 8%	Increased computational cost
	Tran et al. (2021)	Cryptography	Theoretical proofs	Not tested in the presence of attacks
	Maheshwari et al. (2022)	DP	Leakage = 61.74%	Fairness analysis not presented
Security	Wang et al. (2021)	Cryptography	Perplexity = 260 & 190 (transformer &LSTM)	Not tested in the presence of attacks
	Jebreel et al. (2024)	Reputation-based	ASR = 13.36%	Runtime overhead

Cryptography was most commonly used to enhance both privacy and security. Other defenses were DP-based (privacy) and reputation-based (security). Detected limitations include lack of testing in the presence of attacks, and runtime overhead

performance for textual data. However, further analysis revealed that more layers in the bottom model reduce its performance.

5.3 Network aspects

The papers included in this section emphasize the significance of network-related constraints such as bandwidth, network instability, device capability, number of clients and their dropouts, and present solutions for these constraints. We identify several network-related factors that can affect the performance and effectiveness of models in an FL setting for NLP. We also discuss how research copes with these challenges.

5.3.1 Bandwidth usage

One of the most daunting challenges encountered when an FL architecture is implemented is limited bandwidth (Wang et al. 2019). In FL for NLP, bandwidth utilization can be a problem as large amounts of data need to be transferred between clients and the central server. The non-IID data distribution across clients can lead to a high bandwidth utilization as clients have to transfer more data to the central server. Encryption to protect data privacy also increases bandwidth requirements. Complex models like transformer-based models can be difficult to train on clients with limited bandwidth.

Zhang et al. (2020) proposed a novel adaptive compressed FL (ACFL) algorithm to enable adaptive compression and information sharing according to the network conditions. The algorithm can effectively reduce the communication overhead caused by unreliable networks and dynamic bandwidth. It is based on the Choco-Gossip algorithm, which allows the clients to share information, thereby, reducing the number of messages sent to the server. The results of the experiments showed that the proposed algorithm can effectively address the challenges caused by dynamic bandwidth. The average bandwidth utilization by the proposed system for convolutional NN (CNN, LeCun et al. (1989)) was over 93%.

5.3.2 Communication cost

The amount of data that needs to be exchanged between clients and the central server can significantly impact the training time and overall cost of the system. Limited bandwidth can also be an issue as it limits the amount of data that can be transmitted at once, leading to slower training times and increased communication costs.

Zhang et al. (2020) also designed the Cecilia architecture to address communication challenges in FL. Cecilia incorporates the ACFL algorithm, to dynamically adjust the compression rate of shared models based on real-time network conditions. Notably, larger models like CNNs, with high data requirements, benefited most from this adaptive compression and achieved an average compression rate of 54%. In contrast, smaller models, e.g., Bag-Log-Reg or LSTM, were less impacted but still experienced some reduction in communication overhead. Their study did not analyze the impact of their framework on privacy.

Tran et al. (2021) mitigated the problem of communication cost by presenting the ESSP method, which enables a large group of parties to calculate a sum of private inputs jointly. The paper claims their framework reduced the number of training rounds by 5, compared to Downpour SGD (Dean et al. 2012), to achieve the baseline accuracy. Overall, they achieved a promising accuracy of 92% after 100 communication rounds.

FedMask by Li et al. (2021) tackled communication efficiency through the use of heterogeneous masking. FedMask addressed this challenge by allowing each device to learn a personalized and structured sparse deep NN model (Rumelhart et al. 1986a), which could run efficiently on devices. To achieve this, each device learned a sparse binary mask (i.e., 1 bit per network parameter) while keeping the parameters of each local model unchanged. Only these binary masks were communicated between the server and the devices. Instead of learning a shared global model as in classic FL, each device obtained a personalized and structured sparse model that was composed by applying the learned binary mask to the fixed parameters of the local model. The paper claims that FedMask reduced the communication cost by 3.36 times compared to top-k. This technique is impacted by imbalanced data.

Wang et al. (2022) further reduced the communication cost of split learning in a federated setting while minimally influencing the performance vs communication tradeoff. Their FedLite approach compresses the extra communication with a new clustering method and a gradient method of correction. For FL, data inspection is important for several reasons, and automating this process has its benefits because modelers only have access to aggregated outputs (for instance, parameters or metrics). The authors were able to achieve a 51-fold reduction in the communication cost with respect to the system without their compression, at the cost of a 5% accuracy drop.

Finally, Li et al. (2022a) proposed AdaDPS, which explores the impact of leveraging non-sensitive side information and noise addition on communication efficiency in FL. Their experiments showed that AdaDPS requires less noise to guarantee the same privacy. They compared their results with both centralized settings and state-of-the-art FL frameworks. In an FL setting, numerous clients communicate their local updates to a model manager. However, the cost of communicating entire models is not tolerable, especially in resource-constrained environments. Their experiments revealed that AdaDPS achieved a 5% better test accuracy than FedAvg and FedAdam. However, compared to the centralized setting the accuracy was extremely low.

5.3.3 Clients' dropout

When some clients stop participating in the training process, data imbalance, data scarcity or reduced data diversity may ensue, and the model accuracy may decrease. This can result in poor model performance and biased predictions. Since in an FL environment, some clients might inevitably drop out, frameworks must be made immune to it.

Wang et al. (2021) proposed the SEFL framework as a solution to this issue. SEFL eliminates the need for a trusted aggregator, and it was evaluated on NLP tasks. The results demonstrated that it achieved accuracy comparable to existing FL solutions even when clients dropped out. The framework was found to be resilient to client dropouts. With a 75% of clients dropping out, the perplexities increased by 17.41 and 46.34 for LSTM and transformers, respectively.

5.3.4 Computation cost

In FL for NLP, the computation cost can be a problem because large model sizes and high-dimensional data can require significant resources for training, and non-IID data distribution across clients can lead to a high computation cost. Encryption to protect data privacy also increases computation costs and complex models like transformer-based models can be difficult to train on clients with limited resources.

Moreover, another challenge of FL is to balance the trade-off between communication and computation cost, as devices with limited resources such as mobile phones have limited communication bandwidth. The FedMask method presented by Li et al. (2021) maintained this balance well as they also improved computation cost. They achieved an improvement of 1.23 times over baseline techniques.

Dudziak et al. (2022) presented FedorAS, which addresses the computational cost issue by performing resource-efficient federated neural architecture search (NAS). It aims to discover and train promising architectures while minimizing the use of resources. This is achieved by effectively sharing the weights across different devices and by training in a flexible, resource-aware manner. The authors of the paper claim that FedorAS offers significantly lower overhead compared to existing federated NAS techniques, while achieving state-of-the-art performance compared to heterogeneous FL solutions. They provided experimental evidence to support their claims, by evaluating FedorAS across different settings, spanning three different modalities (including text), and comparing its performance with state-of-the-art federated solutions. The results of their experiments showed that FedorAS was able to achieve high performance while maintaining resource efficiency. They made clusters based on the computational capabilities of devices (ranging from 7 to 24 mega floating-point operations per second), and their best perplexity was 3.38. Further research is required to assess the impact of this framework on privacy.

In summary, FL faces several network-related challenges as depicted in Table 8, which slow down the training process and deteriorate the overall performance of the system. Research has been conducted to come up with techniques to overcome these issues. However, there is a lot of room for improvement in all these network aspects of the FL environment when used for NLP tasks.

5.4 Influence of data

Data influence is another factor that must be considered since it originates in the decentralized nature of FL and the heterogeneity of clients. Unlike centralized ML, which is more controlled, FL critically suffers from data-related issues such as non-IID data distribution, class imbalance, and data preprocessing. Non-IID means that the data on each device may have a different distribution of features and labels, which can lead to significant challenges when trying to train a model across all devices. Class imbalance is another issue that is more prevalent in FL. In FL, different devices may have different proportions of data for different classes, which can result in imbalanced class distributions across the entire dataset. Regarding data preprocessing, it must be performed on each device before transmitting updates to the central server for aggregation. This means that each device may preprocess the data differently, which can result in inconsistencies in the data and affect the accuracy of the model.

5.4.1 Non-IID data distribution

In FL, clients may have different data distributions, which can lead to non-IID data. This can affect the overall accuracy and robustness of the model.

To address this problem, Tran et al. (2021) used SDTF for privacy-preserving deep learning models. The main feature of the proposed framework was its ability to work in a

Table 8 Impact of network aspects on FL for NLP tasks

Aim	References	Evaluation	Side effects/limitations
Bandwidth utilization	Zhang et al. (2020)	Avg. utilization = 93.12%	Impact on privacy was not considered
	Zhang et al. (2020)	Compression = 54%	Impact on privacy was not considered
	Tran et al. (2021)	92% acc. after 100 rounds	Performance is lower than the centralized setting and impacted by non-IID data
	Li et al. (2021)	3.36 times improvement in top-k	Data imbalance
Communication cost	Wang et al. (2022)	51 times reduction in comm. cost	5% loss in accuracy
	Li et al. (2022a)	Test Acc. = 17% in (40 rounds)	Cannot keep up with centralize setting
Clients' Dropout	Wang et al. (2021)	Perplexity at 0%. Dropout = 177.41 & 263.78. Perplexity at 75%. Dropout = 194.82 & 310.12	Influenced by dropout
Computation cost	Li et al. (2021)	Computation costs = 58%	Led to privacy loss
	Dudziak et al. (2022)	Best Perplexity = 3.38	Privacy might be impacted

Various studies with their evaluations and limitations are presented. The trade-offs between efficiency and performance, and the need for further research to address privacy and data imbalance issues in FL for NLP are highlighted

decentralized network setting without the need for a trusted third-party server, while simultaneously ensuring the privacy of local data with a low cost of communication bandwidth. The framework used a combination of randomization techniques, secure sum protocols, and secure model sharing protocols to protect the privacy of local data while training deep learning models. The privacy of the approach was theoretically assessed, and the communication cost was experimentally evaluated on balanced and unbalanced class datasets such as UCI SMS Spam. The experiments demonstrated that the proposed approach could obtain high accuracy and was robust to the heterogeneity of decentralized network and non-IID data distributions. The proposed approach achieved both privacy and efficiency while retaining higher model utility than the DP approaches.

Li et al. (2022b) advanced the existing research by using BERT models, as non-IID data can cause performance loss when BERT models are trained in the FL setting. They proposed a framework, FedSplitBERT, which handles heterogeneous data by splitting the BERT encoder layers into a local part and a global part. The local part parameters were trained by the local client only, while the global part parameters were trained by aggregating gradients of multiple clients. Additionally, the paper also explored a quantization method to further reduce communication costs with minimal performance loss. The proposed framework, FedSplitBERT, outperformed baseline methods by a significant margin while reducing the communication cost and being compatible with many existing FL algorithms.

Lastly, Mills et al. (2023) proposed a novel approach for incorporating adaptive optimization techniques into FL with the federated global biased optimizer (FedGBO) algorithm. FedGBO accelerates FL by employing a set of globally biased optimizer values during the client training phase, which helps reduce 'client drift' from non-IID data, while also benefiting from adaptive optimization. The paper also showed that the FedGBO update with a generic optimizer can be reformulated as centralized training using biased gradients and optimizer updates. Furthermore, the paper conducted extensive experiments using 4 realistic FL benchmark datasets and 3 popular adaptive optimizers to compare the performance of state-of-the-art adaptive FL algorithms, and it provided practical insights into the trade-offs associated with the different adaptive-FL algorithms and optimizers for real-world FL deployments.

5.4.2 Class imbalance

Class imbalance refers to the situation in which the number of samples of one class is significantly different from the number of samples of another class in the dataset. This can lead to poor performance of models trained on such datasets, as the models may be biased towards the more frequently occurring class.

Fu et al. (2021) tackled the problem of class imbalance in FL for NLP by proposing a new method called CIC-FL (class imbalance-aware clustered FL). The authors of the paper highlighted that the class imbalance issue in FL was caused by the difference between conditional and joint distributions and that the existing methods that group clients with similar conditional distributions into the same cluster may fail in case of class imbalance. They proposed CIC-FL as a solution to this problem. CIC-FL iteratively bipartitions clients by leveraging a particular feature called LEGLD (locally estimated global label distribution) which is sensitive to concept shift but robust to class imbalance. The authors also showed that CIC-FL was privacy-preserving and communication-efficient. They tested CIC-FL on

benchmark datasets including IMDB, and the results showed that CIC-FL outperformed state-of-the-art clustering methods in FL in the presence of class imbalance.

Another attempt to overcome the class imbalance challenge in FL for NLP is by Duan et al. (2021). The paper addressed the class imbalance issue caused by the FL's non-IID and imbalanced training data distributed in the federated network. The stated problem increased the divergences between the local and global models, further degrading performance. The authors proposed a novel clustered FL framework called FedGroup that groups the training of clients based on the similarities between the clients' optimization directions for high training performance. They constructed a new data-driven distance measure to improve the efficiency of the client clustering procedure. They also implemented a newcomer device cold-start mechanism based on the auxiliary global model for framework scalability and practicality. The paper additionally showed that FedGroup can be combined with the FL optimizer FedProx, and analyzed the convergence and complexity to demonstrate the efficiency of the proposed framework. The results illustrated that FedGroup can significantly improve absolute test accuracy by +3.4% on Sentiment 140 compared to FedProx.

5.4.3 Data preprocessing

Textual data often require a significant amount of preprocessing and cleaning before they can be used for training models. This can add to the overall computation cost and time required for training. Dataset distillation can be considered a preprocessing step in FL, as it is a technique used to prepare the data before they are used in FL model training. The purpose of dataset distillation is to create a smaller and more manageable dataset from the larger, decentralized data sources in an FL setting. This smaller dataset, called a distilled dataset, is used to train a teacher model, which is then used to transfer knowledge to the student models in the FL setting. By distilling the dataset in this way, one improves the communication and computational efficiency of FL, as well as the accuracy of the student models.

Sucholutsky and Schonlau (2021) used dataset distillation as a method to reduce the influence of the dataset on the performance of FL for NLP. They proposed a new algorithm that simultaneously distilled textual data and their labels, thus assigning each synthetic sample a 'soft' label (a distribution of labels) which was previously used only for images. The paper extended the dataset distillation algorithm to distill text data and demonstrated that text distillation outperforms other methods across multiple datasets. For example, models attained almost their original accuracy on the IMDB sentiment analysis task using just 20 distilled sentences. The authors also discussed the limitations of dataset distillation, such as the initialization of distilled labels, the pre-specified number of distilled samples, etc. They also stated the potential benefits of distilled datasets in the FL setting, such as faster training, reduced training times by multiple orders of magnitude, and more.

Table 9 presents a summary of the factors creating data influence and the list of papers that address these issues. The most severe reason for performance degradation in FL for NLP is the non-IID data distribution.

Non-IID data are more vulnerable to privacy attacks in FL because they can potentially reveal information about the underlying data distribution and the clients contributing to the training process. This can lead to privacy breaches such as membership inference attacks (Yang et al. 2019). Another issue with non-IID data is that anti-poisoning techniques can lead to discrimination against minority groups that have legitimately diverse data, resulting

in unfairness and suboptimal models. Singh et al. (2023) aimed to balance the fight against poisoning attacks with the need to accommodate diversity to develop fairer and less discriminatory FL models. The goal was to prevent the exclusion of diverse clients while still detecting poisoning attacks.

6 Codebase accessibility of the surveyed papers

Open access to research codebases significantly facilitates scientific progress by enhancing reproducibility and transparency. Table 10 summarizes whether the surveyed papers provided functional links (non-functional links were discarded) to their codebases, adequate documentation, and information on the libraries used. We found that only 12 out of 36 works offered accessible codebases with comprehensive documentation. PyTorch and TensorFlow were used as the main frameworks.

7 Open challenges

The last research question RQ5 that this SLR aims to answer are the open challenges in FL for NLP. We deal with this question here:

1. *Trade-offs:*

- *Privacy vs performance:* Privacy is one of the reasons for the existence of FL, which aims to keep potentially sensitive local data private to each client. However, improving the privacy of data can compromise accuracy, because the most common solution for privacy in FL (DP) causes a significant loss of accuracy. DP-based ML implementations are motivated by the goal of preserving the accuracy of the learned models. However, they are too loose and lack privacy guarantees that can be established in advance. Instead, they add noise to updates, and Domingo-Ferrer et al. (2021) argue that model accuracy is crucially impacted by this distortion. Blanco-Justicia et al. (2022) experimented and showed that the practical accuracy/privacy tradeoff of DP-ML is worse than that of standard methods used in ML to mitigate overfitting.
- *Communication cost vs model performance:* in FL of NLP the communication cost can be significant because text data can be voluminous. Pre-trained language models are usually large, and sending them to the clients for further training may entail significant communication overheads. Compression techniques are an option

Table 9 Data influence factors in FL for NLP tasks

Factor	References
Non-IID data distribution	Tran et al. (2021), Li et al. (2022b), Mills et al. (2023)
Class imbalance	Duan et al. (2021), Fu et al. (2021)
Data preprocessing	Sucholutsky and Schonlau (2021)

Non-IID data distribution stands out as the most severe reason for performance degradation

Table 10 Reproducibility assessment of the surveyed papers, indicating: (i) whether the implementations were made publicly available; (ii) whether the code was accompanied by detailed documentation; and (iii) the technology or libraries used in each implementation

References	Implementation available	Documentation	Tech. or lib.
Yang et al. (2018)	×	×	TensorFlow
Hard et al. (2018)	https://github.com/steph-jung/GenieType	✓	PyTorch
Ramaswamy et al. (2019)	×	×	TensorFlow lite
Augenstein et al. (2020)	×	×	TensorFlow federated
Zhang et al. (2020)	×	×	×
Huang et al. (2020)	https://github.com/Hazelsuko07/TextHide	✓	TensorFlow
Zawad et al. (2021)	×	×	TensorFlow Federated
Stremmel and Singh (2021)	https://github.com/federated-learning-experiments/fl-text-models	✓	Tensor-Flow, PyTorch
Liu et al. (2021b)	×	×	PyTorch
Li et al. (2021)	×	×	×
Florea et al. (2021)	×	×	PyTorch
Tran et al. (2021)	×	×	Tensor-Flow, Keras
Fu et al. (2021)	×	×	×
Duan et al. (2021)	×	✓	PyTorch
Qin et al. (2021)	×	×	PyTorch
Yuan et al. (2021)	×	×	×
Deng et al. (2021)	×	×	PyTorch
Wang et al. (2021)	×	×	PyTorch
Sucholutsky and Schon-lau (2021)	https://github.com/ilia10000/dataset-distillation	✓	PyTorch
Bhardwaj et al. (2022)	https://github.com/declare-lab/KNOT	✓	PyTorch
Lin et al. (2022)	×	✓	PyTorch
Dudziak et al. (2022)	https://github.com/SamsungLabs/FedorAS	✓	PyTorch
Li et al. (2022b)	×	×	×
Si et al. (2022)	×	×	×

Table 10 (continued)

References	Implementation available	Documentation	Tech. or lib.
Maheshwari et al. (2022)	https://github.com/saist1993/DPNLP	✓	PyTorch
Gupta et al. (2022)	https://github.com/Princeton-SysML/FILM	✓	PyTorch
Yoo and Kwak (2022)	×	×	×
Zhang et al. (2022)	×	×	×
Wang et al. (2022)	×	×	TensorFlow Federated
Li et al. (2022a)	https://github.com/litian96/AdaDPS	✓	TensorFlow
Dong et al. (2022)	×	×	TensorFlow Federated
Cai et al. (2023)	×	×	×
Fowl et al. (2023)	https://github.com/JonasGeiping/breaching	✓	PyTorch
Cai et al. (2023)	×	×	TensorFlow Federated
Mills et al. (2023)	https://github.com/JedMills/FedGBO	✓	PyTorch
Jebreel et al. (2024)	https://github.com/najeebjebreel/FFL	✓	PyTorch

to reduce communication costs, but they often result in a trade-off between model performance and compression rate (Zhang et al. 2020).

- *Time taken to converge vs required accuracy:* the high dimensionality of text data, often characterized by vast vocabularies, increases the time required by model convergence in FL for NLP. This involves a trade-off between achieving high accuracy and ensuring efficient training, particularly in time-critical NLP applications (Cai et al. 2023).

2. *Data heterogeneity:* the textual data used in language models can be non-IID, which makes the models vulnerable to attacks such as local attack timing. Synthetic data generated by differentially private federated generative models can help address this challenge by generating representative data and improving the performance of FL models (Augenstein et al. 2020).
3. *Data preprocessing:* one challenge in FL-based NLP is the preprocessing of textual data, which is crucial for achieving accurate and reliable models. Dataset distillation can be a solution to address this challenge. It involves identifying the most informative data points and using them to construct a distilled dataset, which is smaller in size and easier to process. This approach can improve the efficiency of the FL process and reduce the communication overhead between clients and the aggregator (Sucholutsky and Schonlau 2021). Other preprocessing steps involve sentence splitting, tokenization, and chunking (Sun et al. 2017).

4. *Resource constraints:* client devices may have resource constraints such as limited computational power or storage, which pose a challenge for FL models in NLP. Techniques such as model compression, sparsification, and quantization can be used to reduce the model's size and improve communication efficiency (Zhang et al. 2020).
5. *Network heterogeneity:* network heterogeneity, including different dropout rates, impacts the training effectiveness and stability of FL algorithms, particularly in NLP tasks. Unreliable networks and dynamic bandwidth affect communication compression and adaptation, which influence transmission and aggregation, thereby affecting the overall FL setting (Zhang et al. 2020).

8 Conclusion

This SLR has comprehensively examined FL for NLP tasks from multiple dimensions, unlike previous surveys and reviews that focused solely on model improvement or security and privacy aspects of FL. Moreover, it is the only review that follows a systematic approach to present newer challenges and insights that have emerged as the research in FL for NLP has progressed. All this makes it a valuable resource for researchers and practitioners seeking to advance the field.

Through a comprehensive analysis of recent research, we have identified that considerable effort has been devoted to addressing the challenges of achieving private and efficient FL for NLP tasks. Our analysis covers four key factors that impact the effectiveness of FL-based NLP: privacy and security, model performance, network aspects, and the influence of the data distribution. Each of these factors has been thoroughly examined against the state of the art. We have observed that, to breach the security of FL for NLP, backdooring methods have been used, whereas attacks on privacy exploit gradients, honest-but-curious servers, and eavesdropping. On the other hand, cryptographic methods have been found to be the most frequently used for protecting against privacy and security attacks. Moreover, dealing with textual data in a federated setting is more difficult because it poses extra challenges due to the non-IID nature of text. We have finally presented the current open challenges in FL for NLP.

Acknowledgements This research was funded by the the European Commission (Project H2020-871042 "SoBigData++"), the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and to D. Sánchez), MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe" under grants PID2021-123637NB-I00 "CURLING", and European Union NextGenerationEU/PRTR via INCIBE (project "HERMES" and INCIBE-URV cybersecurity chair). Younas Khan acknowledges financial support from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No. 945413 and Universitat Rovira i Virgili. This paper reflects only the authors' view and the European Research Executive Agency is not responsible for any use that may be made of the information it contains.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material.

If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

AbdulRahman S, Tout H, Ould-Slimane H, Mourad A, Talhi C, Guizani M (2020) A survey on federated learning: the journey from centralized to distributed on-site learning and beyond. IEEE Internet Things J 8(7):5476–5497

Ait-Mlouk A, Alawadi SA, Toor S, Hellander A (2022) Fedqas: privacy-aware machine reading comprehension with federated learning. Appl Sci 12(6):3130

Aledhari M, Razzak R, Parizi RM, Saeed F (2020) Federated learning: a survey on enabling technologies, protocols, and applications. IEEE Access 8:140699–140725

Aljaafari N, Nazzal M, Sawalmeh AH, Khreishah A, Anan M, Algosaibi A, Alnaeem MA, Aldalbahi A, Alhumam A, Vizcarra CP (2022). Investigating the factors impacting adversarial attack and defense performances in federated learning. IEEE Trans Eng Manag Early Access 1–14

Almeida TA, Hidalgo JMG, Yamakami A (2011) Contributions to the study of SMS spam filtering: new collection and results. In: Proceedings of the 11th ACM symposium on document engineering, pp 259–262

Andreina S, Marson GA, Möllering H, Karame G (2021) Baffle: backdoor detection via feedback-based federated learning. In: 2021 IEEE 41st international conference on distributed computing systems (ICDCS). IEEE, pp 852–863

Augenstein S, McMahan HB, Ramage D, Ramaswamy S, Kairouz P, Chen M, Mathews R, y Arcas BA (2020) Generative models for effective ML on private, decentralized datasets. In: 8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020. OpenReview.net

Banabilah S, Aloqaily M, Alsayed E, Malik N, Jararweh Y (2022) Federated learning review: Fundamentals, enabling technologies, and future applications. Inf Process Manag 59(6):103061

Bhardwaj R, Vaidya T, Poria S (2022) KNOT: knowledge distillation using optimal transport for solving NLP tasks. In: Calzolari N, Huang C, Kim H, Pustejovsky J, Wanner L, Choi K, Ryu P, Chen H, Donatelli L, Ji H, Kurohashi S, Paggio P, Xue N, Kim S, Hahm Y, He Z, Lee TK, Santus E, Bond F, Na S (eds) Proceedings of the 29th international conference on computational linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12–17, 2022. International Committee on Computational Linguistics, pp 4801–4820

Biggio B, Nelson B, Laskov P (2012) Poisoning attacks against support vector machines, pp 1–8. arXiv preprint arXiv:1206.6389

Blanco-Justicia A, Domingo-Ferrer J, Martínez S, Sánchez D, Flanagan A, Tan KE (2021) Achieving security and privacy in federated learning systems: survey, research challenges and future directions. Eng Appl Artif Intell 106:104468

Blanco-Justicia A, Sánchez D, Domingo-Ferrer J, Muralidhar K (2022) A critical review on the use (and misuse) of differential privacy in machine learning. ACM Comput Surv 55(8):1–16

Blodgett SL, Green L, O'Connor B (2016) Demographic dialectal variation in social media: a case study of African–American English. arXiv preprint arXiv:1608.08868

Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp 1175–1191

Cai S, Chai D, Yang L, Zhang J, Jin Y, Wang L, Guo K, Chen K (2023) Secure forward aggregation for vertical federated neural networks, pp 115–129. https://doi.org/10.1007/978-3-031-28996-5_9

Cai D, Wu Y, Wang S, Lin FX, Xu M (2023) Autofednlp: an efficient fednlp framework, pp 1–13. arXiv preprint arXiv:2205.10162

Caldas S, Duddu SMK, Wu P, Li T, Konečnỳ J, McMahan HB, Smith V, Talwalkar A (2018) Leaf: a benchmark for federated settings. arXiv preprint arXiv:1812.01097

Chen Y, Qin X, Wang J, Yu C, Gao W (2020) Fedhealth: a federated transfer learning framework for wearable healthcare. IEEE Intell Syst 35(4):83–93

Chen Y, Ning Y, Slawski M, Rangwala H (2020) Asynchronous online federated learning for edge devices with non-iid data. In: 2020 IEEE international conference on big data (big data). IEEE, pp 15–24

Chhikara P, Singh P, Tekchandani R, Kumar N, Guizani M (2020) Federated learning meets human emotions: a decentralized framework for human-computer interaction for iot applications. IEEE Internet Things J 8(8):6949–6962

Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation. BMC Genom 21:1–13
Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, Ranzato M, Senior A, Tucker P, Yang K, et al. (2012) Large scale distributed deep networks. Adv Neural Inf Process Syst 25. In: Pereira F, Burges CJ, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/6aca97005c68f1206823815f66102863-Paper.pdf
Deng J, Wang Y, Li J, Shang C, Liu H, Rajasekaran S, Ding C (2021) Tag: Gradient attack on transformer-based language models, pp 1–11. arXiv preprint arXiv:2103.06819
Devlin J, Chang MW, Lee K, Toutanova K (2018). Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
Domingo-Ferrer J, Sánchez D, Blanco-Justicia A (2021) The limits of differential privacy (and its misuse in data release and machine learning). Commun ACM 64(7):33–35
Dong W, Wu X, Li J, Wu S, Bian C, Xiong D (2022). Fewfedweight: few-shot federated learning framework across multiple NLP tasks. arXiv preprint arXiv:2212.08354
Duan M, Liu D, Ji X, Liu R, Liang L, Chen X, Tan Y (2021) Fedgroup: efficient federated learning via decomposed similarity-based clustering. In: 2021 IEEE intl conf on parallel & distributed processing with applications, big data & cloud computing, sustainable computing & communications, social computing & networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, pp 228–237
Dudziak L, Laskaridis S, Fernandez-Marques J (2022) Fedoras: federated architecture search under system heterogeneity, pp 1–28. arXiv preprint arXiv:2206.11239
Fang M, Cao X, Jia J, Gong NZ (2020) Local model poisoning attacks to byzantine-robust federated learning. In: Proceedings of the 29th USENIX conference on security symposium, pp 1623–1640
Florea IM, Constantin M, Ciocîrlan SD (2021) Benchmarking privacy in text classification. In: 2021 20th RoEduNet conference: networking in education and research (RoEduNet). IEEE, pp 1–6
Fowl L, Geiping J, Reich S, Wen Y, Czaja W, Goldblum M, Goldstein T (2023). Decepticons: corrupted transformers breach privacy in federated learning for language models, pp 1–26. arXiv preprint arXiv:2201.12675
Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pp 1322–1333
Fu Y, Liu X, Tang S, Niu J, Huang Z (2021) Cic-fl: enabling class imbalance-aware clustered federated learning over shifted distributions. In: Database systems for advanced applications: 26th international conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part I 26. Springer, pp 37–52
Fung C, Yoon CJ, Beschastnikh I (2020) The limitations of federated learning in sybil settings. In: RAID, pp 301–316
Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Proj Rep, Stanford 1(12): 2009
Gosselin R, Vieu L, Loukil F, Benoit A (2022) Privacy and security in federated learning: a survey. Appl Sci 12(19):9901
Gupta S, Huang Y, Zhong Z, Gao T, Li K Chen D (2022) Recovering private text in federated learning of language models, pp 1–18. arXiv preprint arXiv:2205.08514
Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, Eichner S, Kiddon C, Ramage D (2018) Federated learning for mobile keyboard prediction, pp 1–7. arXiv preprint arXiv:1811.03604
Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Huang Y, Song Z, Chen D, Li K, Arora S (2020). Texthide: tackling data privacy in language understanding tasks, pp 1–15. arXiv preprint arXiv:2010.06053
Injadat M, Moubayed A, Nassif AB, Shami A (2021) Machine learning towards intelligent systems: applications, challenges, and opportunities. Artif Intell Rev 54:3299–3348
Jebreel NM, Domingo-Ferrer J, Li Y (2023) Defending against backdoor attacks by layer-wise feature analysis. In: The 26th Pacific-Asia conference on knowledge discovery and data mining (PAKDD 2023). Springer, pp 428–440
Jebreel NM, Domingo-Ferrer J, Blanco-Justicia A, Sánchez D (2024) Enhanced security and privacy via fragmented federated learning. IEEE Trans Neural Netw Learn Syst 35:6703–6717
Jelinek F, Mercer RL, Bahl LR, Baker JK (1977) Perplexity–a measure of the difficulty of speech recognition tasks. J Acoust Soc Am 62(S1):S63–S63
Jiang JC, Kantarci B, Oktug S, Soyata T (2020) Federated learning in smart city sensing: challenges and opportunities. Sensors 20(21):6230

Kanagavelu R, Wei Q, Li Z, Zhang H, Samsudin J, Yang Y, Goh RSM, Wang S (2022) Ce-fed: communication efficient multi-party computation enabled federated learning. Array 15:100207

Kanani P, Marathe VJ, Peterson D, Harpaz R, Bright S (2022). Private cross-silo federated learning for extracting vaccine adverse event mentions. In: Machine learning and principles and practice of knowledge discovery in databases: international workshops of ECML PKDD 2021, virtual event, September 13–17, 2021, proceedings, part II. Springer, pp 490–505

Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: strategies for improving communication efficiency, pp 1–10. arXiv preprint arXiv:1610.05492

Lang K (1995) Newsweeder: learning to filter netnews. In: Machine learning proceedings. Elsevier, pp 331–339

LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackel L (1989) Handwritten digit recognition with a back-propagation network. Adv Neural Inf Process Syst 2. In: Touretzky D (ed) Advances in neural information processing systems, Morgan-Kaufmann. https://proceedings.neurips.cc/paper_files/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf

Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Maga 37(3):50–60

Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020) Federated optimization in heterogeneous networks. Proc Mach Learn Syst 2:429–450

Lim WYB, Luong NC, Hoang DT, Jiao Y, Liang YC, Yang Q, Niyato D, Miao C (2020) Federated learning in mobile edge networks: a comprehensive survey. IEEE Commun Surv Tutor 22(3):2031–2063

Lin BY, He C, Ze Z, Wang H, Hua Y, Dupuy C, Gupta R, Soltanolkotabi M, Ren X, Avestimehr S (2022) Fednlp: benchmarking federated learning methods for natural language processing tasks. In: Carpuat M, de Marneffe M, Ruíz IVM (eds) Findings of the association for computational linguistics: NAACL 2022, Seattle, WA, USA, July 10–15, 2022. Association for Computational Linguistics, pp 157–175

Li Z, Sit S, Wang J, Xiao J (2022) Federated split bert for heterogeneous text classification. In: 2022 international joint conference on neural networks (IJCNN). IEEE, pp 1–8

Li A, Sun J, Zeng X, Zhang M, Li H, Chen Y (2021). Fedmask: joint computation and communication-efficient personalized federated learning via heterogeneous masking. In Proceedings of the 19th ACM conference on embedded networked sensor systems, pp 42–55

Liu Y, Huang A, Luo Y, Huang H, Liu Y, Chen Y, Feng L, Chen T, Yu H, Yang Q (2020) Fedvision: an online visual object detection platform powered by federated learning. Proc AAAI confer Artif Intell 34:13172–13179

Liu M, Ho S, Wang M, Gao L, Jin Y, Zhang H (2021). Federated learning meets natural language processing: a survey. arXiv preprint arXiv:2107.12603

Liu S, Xu S, Yu W, Fu Z, Zhang Y, Marian A (2021) Fedct: federated collaborative transfer for recommendation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 716–725

Li T, Zaheer M, Reddi S, Smith V (2022) Private adaptive optimization with side information. In: International conference on machine learning. PMLR, pp 13086–13105

Long G, Tan Y, Jiang J, Zhang C (2020) Federated learning for open banking. In: Federated learning: privacy and incentive. Springer, pp 240–254

Maas A, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp 142–150

Maheshwari G, Denis P, Keller M, Bellet A (2022) Fair NLP models with differentially private text encoders. In: Goldberg Y, Kozareva Z, Zhang Y (eds) Findings of the association for computational linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7–11, 2022. Association for Computational Linguistics, pp 6913–6930

McMahan HB, Moore E, Ramage D, y Arcas BA. (2016) Federated learning of deep networks using model averaging. arXiv:1602.05629

McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR, pp 1273–1282

Mills J, Hu J, Min G, Jin R, Zheng S, Wang J (2023) Accelerating federated learning with a global biased optimiser. IEEE Trans Comput 72:1804–1814. https://doi.org/10.1109/TC.2022.3212631

Moher, D., A. Liberati, J. Tetzlaff, D.G. Altman, and t. PRISMA Group* (2009) Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. Ann Int Med 151(4):264–269

Morgan SP, Teachman JD (1988) Logistic regression: description, examples, and comparisons. J Marriage Fam 50(4):929–936

Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G (2021) A survey on security and privacy of federated learning. Fut Gener Comput Syst 115:619–640

Murphy KP (2012) Machine learning: a probabilistic perspective. MIT Press

Ni J, Li J, McAuley J (2019) Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 188–197

Nishio T, Yonetani R (2019) Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019–2019 IEEE international conference on communications (ICC). IEEE, pp 1–7

Oktay H, Taylor BJ, Jensen DD (2010) Causal discovery in social media using quasi-experimental designs. In: Proceedings of the 1st workshop on social media analytics, pp 1–9

Prabhu OS, Gupta PK, Shashank P, Chandrasekaran K, Usha D (2021) Towards a federated learning approach for nlp applications. In: Applications of artificial intelligence and machine learning: select proceedings of ICAAAIML 2020. Springer, pp 157–167

Qammar A, Ding J, Ning H (2022) Federated learning attack surface: taxonomy, cyber defences, challenges, and future directions. Artif Intell Rev 55(5):3569–3606. https://doi.org/10.1007/s10462-021-10098-w

Qin H, Chen G, Tian Y, Song Y (2021) Improving federated learning for aspect-based sentiment analysis via topic memories. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 3942–3954

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I et al (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9

Ramaswamy S, Mathews R, Rao K, Beaufays F (2019) Federated learning for emoji prediction in a mobile keyboard, pp 1–5. arXiv preprint arXiv:1906.04329

Reddi S, Charles Z, Zaheer M, Garrett Z, Rush K, Konečnỳ J, Kumar S, McMahan HB (2020) Adaptive federated optimization. arXiv preprint arXiv:2003.00295

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. Biometrika 71:599–607

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536

Shinde SS, Bozorgchenani A, Tarchi D, Ni Q (2021) On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems. IEEE Trans Veh Technol 71(2):2041–2057

Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). IEEE, pp 3–18

Shyu CR, Putra KT, Chen HC, Tsai YY, Hossain KT, Jiang W, Shae ZY (2021) A systematic review of federated learning in the healthcare area: from the perspective of data properties and applications. Appl Sci 11(23):11191

Singh AK, Blanco-Justicia A, Domingo-Ferrer J (2023) Fair detection of poisoning attacks in federated learning on non-iid data. Data Min Knowl Discov 37(5):1998–2023. https://link.springer.com/article/10.1007/s10618-022-00912-6

Si S, Wang J, Zhang R, Su Q, Xiao J (2022) Federated non-negative matrix factorization for short texts topic modeling with mutual information. In: 2022 international joint conference on neural networks (IJCNN). IEEE, pp 1–7

Soltani B, Haghighi V, Mahmood A, Sheng QZ, Yao L (2022). A survey on participant selection for federated learning in mobile networks. In: Proceedings of the 17th ACM workshop on mobility in the evolving internet architecture, pp 19–24

Sousa S, Kern R (2023) How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing. Artif Intell Rev 56(2):1427–1492. https://doi.org/10.1007/s10462-022-10204-6

Stremmel J, Singh A (2021) Pretraining federated text models for next word prediction. In: Advances in information and communication: proceedings of the 2021 future of information and communication conference (FICC), vol 2. Springer, pp 477–488

Sucholutsky I, Schonlau M (2021). Soft-label dataset distillation and text dataset distillation. In: 2021 international joint conference on neural networks (IJCNN). IEEE, pp 1–8

Sun S, Luo C, Chen J (2017) A review of natural language processing techniques for opinion mining systems. Inf Fus 36:10–25

Sun G, Cong Y, Dong J, Wang Q, Lyu L, Liu J (2021) Data poisoning attacks on federated machine learning. IEEE Internet Things J 9(13):11365–11375

Tolpegin V, Truex S, Gursoy ME, Liu L (2020). Data poisoning attacks against federated learning systems. In: Computer security–ESORICS 2020: 25th European symposium on research in computer security, ESORICS 2020, Guildford, UK, September 14–18, 2020, proceedings, part I, vol 25. Springer, pp 480–501

Tran AT, Luong TD, Karnjana J, Huynh VN (2021) An efficient approach for privacy preserving decentralized deep learning models based on secure multi-party computation. Neurocomputing 422:245–262
Wang S, Tuor T, Salonidis T, Leung KK, Makaya C, He T, Chan K (2019) Adaptive federated learning in resource constrained edge computing systems. IEEE J Sel Areas Commun 37(6):1205–1221
Wang X, Chen W, Xia J, Wen Z, Zhu R, Schreck T (2022) Hetvis: a visual analysis approach for identifying data heterogeneity in horizontal federated learning. IEEE Trans Vis Comput Graph 29(1):310–319
Wang C, Deng J, Meng X, Wang Y, Li J, Lin S, Han S, Miao F, Rajasekaran S, Ding C (2021) A secure and efficient federated learning framework for NLP. In: Moens M, Huang X, Specia L, Yih SW (eds) Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 7–11 November, 2021. Association for Computational Linguistics, pp 7676–7682
Wang J, Qi H, Rawat AS, Reddi S, Waghmare S, Yu FX, Joshi G (2022) Fedlite: a scalable approach for federated learning on resource-constrained clients, pp 1–17. arXiv preprint arXiv:2201.11865
Warstadt A, Singh A, Bowman SR (2019) Neural network acceptability judgments. Trans Assoc Comput Linguist 7:625–641
Wu C, Wu F, Lyu L, Huang Y, Xie X (2022) Communication-efficient federated learning via knowledge distillation. Nat Commun 13(1):2032
Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F (2021) Federated learning for healthcare informatics. J Healthc Informat Res 5:1–19
Yan N, Wang K, Pan C, Chai KK (2022) Private federated learning with misaligned power allocation via over-the-air computation. IEEE Commun Lett 26(9):1994–1998
Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H (2019) Federated learning, synthesis lectures on artificial intelligence and machine. Learning 13(3):1–207
Yang T, Andrew G, Eichner H, Sun H, Li W, Kong N, Ramage D, Beaufays F (2018) Applied federated learning: improving google keyboard query suggestions. arXiv preprint arXiv:1812.02903
Yoo K, Kwak N (2022) Backdoor attacks in federated learning by rare embeddings and gradient ensembling. In: Goldberg Y, Kozareva Z, Zhang Y (eds) Proceedings of the 2022 conference on empirical methods in natural language processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7–11, 2022. Association for Computational Linguistics, pp 72–88
Yuan X, Ma X, Zhang L, Fang Y, Wu D (2021) Beyond class-level privacy leakage: breaking record-level privacy in federated learning. IEEE Internet Things J 9(4):2555–2565
Zawad S, Ali A, Chen PY, Anwar A, Zhou Y, Baracaldo N, Tian Y, Yan F (2021) Curse or redemption? how data heterogeneity affects the robustness of federated learning. Proc AAAI Confer Artif Intell 35:10807–10814
Zhang X, Zhu X, Wang J, Yan H, Chen H, Bao W (2020) Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks. Inf Sci 540:242–262
Zhang S, Yin H, Chen T, Huang Z, Nguyen QVH, Cui L (2022) Pipattack: Poisoning federated recommender systems for manipulating item promotion. In: Proceedings of the 15th ACM international conference on web search and data mining, pp 1415–1423
Zhao L, Xu H, Wang J, Chen Y, Chen X, Wang Z (2022) Computation-communication resource allocation for federated learning system with intelligent reflecting surfaces. Arab J Sci Eng 47:10203–10209
Zhu H, Xu J, Liu S, Jin Y (2021) Federated learning on non-iid data: a survey. Neurocomputing 465:371–390

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.