A Survey on Privacy Risks and Protection in Large Language Models

Kang Chen[1,2], Xiuze Zhou[3], Yuanguo Lin[1], Shibo Feng[4],
Li Shen[5], Pengcheng Wu[6†]

[1]School of Computer Engineering, Jimei University, Xiamen, 361021,
China.
[2]College of Science, Mathematics and Technology, Wenzhou-Kean
University, Wenzhou, 325060, China.
[3]Information Hub, The Hong Kong University of Science and
Technology (Guangzhou), Guangzhou, 511453, China.
[4]College of Computing and Data Science, Nanyang Technological
University, Singapore, 639798, Singapore.
[5]School of Professional Studies, New York University, New York, 10003,
United States.
[6]Webank-NTU Joint Research Institute on Fintech, Nanyang
Technological University, Singapore, 639798, Singapore.

Contributing authors: chenkang@kean.edu;
xz.zhou@connect.hkust-gz.edu.cn; xdlyg@jmu.edu.cn;
shibo001@ntu.edu.sg; ls6743@nyu.edu; pengchengwu@ntu.edu.sg;
[†]These authors contributed equally to this work.

Abstract

Although Large Language Models (LLMs) have become increasingly integral to
diverse applications, their capabilities raise significant privacy concerns. This survey offers a comprehensive overview of privacy risks associated with LLMs and
examines current solutions to mitigate these challenges. First, we analyze privacy leakage and attacks in LLMs, focusing on how these models unintentionally
expose sensitive information through techniques such as model inversion, training data extraction, and membership inference. We investigate the mechanisms
of privacy leakage, including the unauthorized extraction of training data and
the potential exploitation of these vulnerabilities by malicious actors. Next, we

1

review existing privacy protection against such risks, such as inference detection, federated learning, backdoor mitigation, and confidential computing, and assess their effectiveness in preventing privacy leakage. Furthermore, we highlight key practical challenges and propose future research directions to develop secure and privacy-preserving LLMs, emphasizing privacy risk assessment, secure knowledge transfer between models, and interdisciplinary frameworks for privacy governance. Ultimately, this survey aims to establish a roadmap for addressing escalating privacy challenges in the LLMs domain.

Introduction

Large Language Models (LLMs) are powerful tools in Natural Language Processing (NLP), employing deep learning algorithms to interpret and produce text that resembles human language. They have the excellent ability to follow instructions and perform various text-based activities, such as writing and coding [1–3]. In recent years, LLMs have shown great potential in advancing artificial intelligence, which represents a significant leap in the field [4, 5]. They are also recognized as excellent contextual learners [6]. The large-scale adoption of LLMs has introduced a new era of convenient knowledge transfer for many NLP tasks [7].

An LLM exemplified by ChatGPT is widely used for solving various NLP-related tasks in daily personal life [8, 9]. Increasing attention is being paid to the impact of LLMs on privacy. With the continual improvement in the reasoning abilities of LLMs, current research on privacy primarily focuses on the extraction of memory training data [10]. LLMs supplement limited empirical knowledge with domain-specific insights, although the reliability of this generated knowledge remains uncertain. Combining LLMs with input from multiple stakeholders improves knowledge quality and scalability; however, it may also raise privacy concerns [11]. The training data for LLMs is extracted typically from a wide range of Internet texts, which may contain personal, sensitive, or privacy-related information. An undesirable side effect of using the extensive Internet for training is that the model may retain potentially sensitive information, which could be leaked to a third party [10].

Current privacy research on LLMs primarily focuses on the extraction of memory training data [10]. These models automatically store user information from conversations to provide personalized responses. Although this is beneficial, it raises privacy and cybersecurity concerns [12]. The personalized deployment of LLMs in split learning also carries privacy risks, necessitating strong security measures to protect raw data and intermediate representations, particularly in sensitive areas like healthcare [13]. LLMs face challenges during inference and training. The memory of the model stores vast amounts of data, including sensitive information, which can lead to the unintentional generation of content resembling the training data, potentially leaking personal or proprietary details. Additionally, the unpredictability of the output of

the model complicates security, especially in dynamic or multi-round scenarios. The variety of training data sources makes it difficult to assess the sensitivity of each data piece. With continual improvements in LLMs' reasoning, these models can infer personal attributes from text, reaching new levels of capability [10].

In LLMs operations, privacy protection technologies are becoming increasingly important, especially in the digital age, where safeguarding personal and sensitive data is critical. These technologies help legal professionals navigate complex data protection regulations, while improving compliance with data processing and storage requirements. Privacy protection methods, including data cleaning, differential privacy [14], and confidential computing [15], ensure the secure handling of user information, thereby preserving privacy and reducing the risk of accidental data exposure. To maintain user privacy throughout the data lifecycle, a framework for securing Retrieval-Augmented Generation (RAG) pipelines incorporates encryption, zero-trust principles, and guardrails [16]. A conceptual solution has also been proposed to enhance user privacy by detecting and anonymizing sensitive named entity categories, while maintaining context by substituting original entities with functionally equivalent ones [17]. These methods significantly improve the privacy protection of LLMs.

Motivation. The primary motivation for investigating privacy issues in LLMs is to ensure the accuracy and reliability of model outputs. In critical areas such as education, healthcare, and law, incorrect information can lead to misleading conclusions and serious social consequences, such as misdiagnosis or legal errors. Additionally, as society increasingly values privacy protection, safeguarding users' personal information has become essential. The improper use or leakage of sensitive data during training can lead to legal liability and a crisis of trust, negatively impacting both businesses and users. While significant research has been conducted on privacy in machine learning, the specific challenges of LLMs have received insufficient attention. These challenges include privacy management, model complexity, and the practical implementation of privacy protection technologies. This paper aims to support the development of privacy protection in LLMs through systematic review and research, ensuring their security and reliability in real-world applications, thus enhancing user trust and upholding social and ethical standards.

Existing surveys have explored various aspects of the security and privacy of LLMs. For instance, Das et al. [18] provide a broad overview of the challenges and potential defenses; Yao et al. [19] categorize findings into beneficial applications, offensive uses, and inherent vulnerabilities; Esmradi et al. [20] review a wide range of attack techniques, implementation methods, and mitigation strategies in LLMs. These surveys have made valuable contributions to understanding the risks associated with LLMs and the various defenses that can be employed. However, these surveys often address privacy issues independently or without a systematic framework. In contrast, our survey presents a unified classification that integrates privacy concerns. We further classify these issues based on their unique characteristics, going a step further in our analysis. This fine-grained classification approach emphasizes the interconnectedness of these domains. Focusing on privacy highlights its critical importance in protecting user privacy and meeting regulatory standards. By systematically analyzing privacy

Fig. 1 Taxonomy of LLM's privacy in this survey.

concerns, our survey underscores their significance and provides actionable insights for enhancing the ethical use of LLMs.

Contributions. Our work provides an in-depth analysis of the unique challenges faced by LLMs in privacy protection. We studied eleven risks and attacks in privacy, classified them according to their characteristics, and provided definitions and corresponding mitigation techniques for each classification. After critically analyzing the advantages and disadvantages of existing technologies, we explored how to effectively apply these technologies to enhance the security and user privacy protection of LLMs. These contributions not only fill the gap in current research and propose potential improvements or new approaches to privacy protection in the context of LLMs, but also offer valuable guidance for future work.

The remainder of this paper is organized as follows (as illustrated in Figure 1). The architecture and vulnerabilities of LLMs are introduced in Section 2. The privacy threats to LLMs are discussed in Section 3. Popular mitigation techniques for different types of attacks are discussed in Section 4. Future research directions are presented in Section 5. Finally, the conclusion is given in Section 6.

Background

Architecture and vulnerabilities of LLMs

As a deep learning-based NLP model, LLMs have a complex and multi-stage workflow to transform collected data into useful text responses. On one hand, the entire process begins with the collection of the dataset, which includes users' natural language data. Recent studies have shown that the quality of this initial data significantly impacts downstream performance [21, 22]. The data is then preprocessed for conversion into a format compatible with the model, with any irrelevant or redundant information being eliminated to improve quality. In the core stages of pre-training and fine-tuning, the system learns language rules using large-scale text data to develop a broad understanding of language. Subsequently, the model is fine-tuned on specific task data to better align with particular application scenarios or task requirements [23]. Throughout this process, the integrity and quality of the data are vital for both pre-training and fine-tuning.

On the other hand, the process also introduces privacy risks, particularly during the data collection and model deployment phases. The collection of large amounts of textual data, which may contain personal or sensitive information [24], along with the real-time interaction between users and the model, increases the risk of privacy leakage. Sensitive information provided by users may be processed and stored by the model, making it susceptible to exploitation by attackers who can exploit vulnerabilities to access this information. Ultimately, the model deployment phase integrates the trained and fine-tuned models into practical applications. This process is illustrated in Figure 2.

During user interaction with LLMs, when users input sensitive information as part of their prompts [25], the starting point of privacy issues, the first step in the overall process of data provision, commences. Recent research demonstrates that even anonymized prompts can be reverse-engineered to recover private information [26]. During input, users may unintentionally provide personal information, confidential data, or sensitive content. If this information is handled incorrectly, it may lead to privacy leakage or attacks from malicious actors.

LLMs Vulnerabilities

According to recent studies, privacy vulnerabilities in LLMs are complex and profound [14, 19]. These vulnerabilities can be classified into different categories according to their characteristics, including the following: privacy attacks, privacy leakage, contextual leakage [27], and backdoor attacks [18]. The privacy risks discussed in this paper are typically categorized into target-based or method-based approaches. In the domain of LLMs, privacy involves respecting and protecting personal information, while minimizing unnecessary risks to user data.

The vulnerabilities of LLMs, with a focus on privacy concerns, are examined in this paper. Specifically, we examine privacy leakage and three types of attacks targeting the following: models, data, and users themselves. Furthermore, we note that different types of attacks often employ similar methods; for example, the inclusion of data

Fig. 2 LLM Privacy Risks: Data Flow Analysis.

poisoning and backdoor attacks, which manipulate the behavior and output of LLMs by introducing malicious samples into the training data [14, 18]. All existing privacy attack methods in the literature have the potential to compromise LLMs, raising significant privacy concerns.

Privacy Issues of LLMs

When it comes to LLMs, privacy is a significant concern. We have divided privacy issues into two categories based on how attackers can access sensitive information: privacy leakage and privacy attacks. Privacy leakage denotes the exploitation of LLM vulnerabilities by attackers to collect sensitive information; whereas, privacy attacks involve attackers breaching the defenses of LLMs through various methods to obtain this information. The methods of privacy leakage are diverse. A detailed classification of these types is provided in Table 1 and Table 2. Next, we briefly introduce these two types of privacy threats and their impacts.

Privacy Leakage

LLMs pose significant privacy risks that can be categorized based on their characteristics. These risks encompass various forms of data exposure that undermine user confidentiality. Understanding these privacy risks is essential for developing strategies to protect user data and maintain confidentiality.

Sensitive Information Leakage

When interacting with LLMs, users may enter personal sensitive details, including their name, phone number, address, ID card number, and bank account information. Once stored or processed by the model, this information may be improperly used or

Table 1 Overview of Privacy Leakage Categories.

Category	Work	Method	Evaluated Model	Dataset	Evaluation Metric
	[28]	Design probe	GPT-3	/	Performance
	[29]	Zero-shot robustness evaluation	DeBERTa-L, BART-L, etc	SST-2, QQP, MNLI, etc	ASR
Sensitive Information Leakage	[30]	Multi-turn approach	GPT-4, GPT-3, ChatGPT	National Flag-Drawing, etc	ROUGE-1, ChrF++, etc
	[31]	Overlap analysis	Flan-PaLM, Med-PaLM2, etc	MedQA (USMLE), PubMedQA, MedMCQA, etc	Acc
	[32]	Zero-shot, Law Recitation, Direct Prompt, LLM API	MPT-7B, Llama2-7B, etc	GOLDCOIN-HIPAA	Acc, Prec, Rec, etc
Contextual Leakage	[27]	Differential Privacy	GPT-4, ChatGPT, InstructGPT, etc	/	Sensitivity Score, Rate, Error Rate
	[10]	Anonymization Alignment	PaLM2-Chat, GPT-4, etc	Enron-Email, PAN competition, etc	Top-k accuracies, Jaro-Winkler, etc
Personal Preferences Leakage	[33]	Web search	GPT-3.5, GPT-4	/	Acc

Table 2 Overview of Privacy Attack Categories.

Category	Work	Method	Evaluated Model	Dataset	Evaluation Metric
	[34]	Layer weight poisoning training	PTMs	SST-2, IMDB, etc	LFR, Clean Acc
	[35]	Restricted Inner Product Poison Learning	BERT, XLNet	SST-2, OffensEval, etc	LFR, Clean Acc
Backdoor Attacks	[36]	Dynamic Surgery	ResNet-18, etc	IMDB, SST-2	distinct, BLEU, etc
	[8]	Model-Editing Techniques	GPT-2-XL, GPT-J	SST-2,AGNews, etc	ASR,CACC
	[37]	Big machine learning	Softmax, MLP, DAE	FiveThirtyEight, GSS	Correct rate, acc, etc
Model Inversion Attacks	[38]	Generative adversarial network	VGG16, ResNet-152, etc	MNIST, ChestX-ray8, etc	PSNR, Attack Acc, etc
	[39]	Word embedding perturbation	Tiny-BERT, BERT	Emotion/Yelp Dataset	RR, Acc, PLL
Model Stealing Attacks	[40]	Data-free model extraction	Resnet-34-8x, etc	SVHN, CIFAR-10.	Acc
	[41]	Prompt engineering	ChatGPT, LLaMA	RetrievalQA, Alpaca-GPT4	Acc, recall, etc
Data Stealing Attacks	[42]	Fine-Tuning	GPT-3.5-turbo, Mistral-7B	Do, D'o	ASR
	[43]	Spilt learning	LeNet-5, VGG16, etc	MNIST, CIFAR-10, etc	Complexity
Training Data Extraction Attacks	[44]	Special Characters Attack	Llama-2-Chat, etc	/	ASR, Count
	[24]	Proof-of-concept Attack	GPT-2	Top-n, Temperature, Internet	Perplexity, Small, etc
Membership Inference Attacks	[45]	Multiple regularization generation, self-prompt	GPT-2, GPT-J, Falcon-7B, LLaMA-7B	Wikitext-103, XSum, etc	AUC
	[46]	overlap analysis	GPT-2-SMALL, etc	Pile-CC, Wikipedia, etc	AUC, ROC, etc
	[47]	Membership inference	Logistic Regression, etc	Loc-30, Pur-100, etc	AUC, Acc
Attribute Inference Attacks	[48]	Attribute inference	SAN, SBA	Google+	Precision, Recall, F-Score
	[49]	Model Extraction	BERT-based API	/	/

leaked. Below is a classification of sensitive information leakage caused by different methods.

Sensitive Query. Privacy leakage in LLMs often results from users mishandling sensitive information.

For example, when interacting with LLMs, users are advised against disclosing sensitive or personally identifiable information, as doing so may lead to privacy risks [50]. User input can be incorporated into the knowledge base for training these models and improving tools; however, this caution has not prevented some LLM users from including sensitive data in their prompts [25]. Kshetri [25] notes that there is some confusion regarding the nature and degree of risks involved when users include sensitive details in their input.

Some users believe that the information they provide is stored in the ChatGPT database, which could lead to the potential leakage of this data to others in response to different queries [24].

Chat-based interaction with LLMs has proved to be a powerful tool for tasks such as programming, academic writing, and medical diagnosis [28]. However, despite their usefulness, LLMs present significant privacy and security risks. Although user inputs are not automatically added to the training data of a model, they are often stored by LLM operators (e.g., OpenAI) and could be accessed for model development or other purposes [24]. This raises concerns about the potential for sensitive information being exposed inadvertently. Previous work has demonstrated that sensitive queries can result in private information being leaked, either through direct access to model parameters or through adversarial probing of the model [37]. Additionally, although LLMs, such as ChatGPT generate responses based on pre-trained models, which do not inherently merge sensitive information into the model or share it with other users, the risk of leakage remains significant. Studies have shown that even pre-trained models, when exposed to sensitive queries, can unintentionally recall or expose personal data, due to the nature of their training processes and the large-scale datasets [51]. In summary, even though the risks associated with LLMs may not always align with user expectations, they are still significant and must be carefully monitored.

Sensitive Information Exposed by Fine-tuning. Currently, LLMs have achieved significant performance with various NLP tasks [29, 52]. However, when LLMs are applied to specialized fields, they inevitably encounter issues such as hallucination [30, 53], insufficient professional knowledge in specific areas [31], and a failure to integrate the latest knowledge into constantly evolving industry scenarios [4]. Then, using high-quality, domain-specific knowledge, researchers fine-tune specialized LLMs based on powerful general-purpose LLMs. Fine-tuning an LLM is re-training the model by providing additional data from specific domains built upon the pre-trained base model, thereby making it more applicable to particular fields. The purpose of fine-tuning a specific LLM with high-quality knowledge is to improve the performance and accuracy of the model in that domain. By incorporating advanced knowledge and data from particular fields into LLMs, the models better understand and generate text content relevant to those fields, thereby enhancing their applicability and usability. However, when fine-tuning an LLM, it is often necessary to train with domain-specific datasets that may contain personal sensitive information, including personally identifying information and health records [54]. If the data is not properly processed, desensitized, or encrypted, the model may learn patterns related to sensitive information during training, potentially leading to sensitive information exposure.

Contextual Leakage

Privacy is not a standalone concept confined to conventional confidential information (such as identification numbers); instead, it is closely connected to complex societal frameworks, which makes identifying and analyzing potential privacy violations more challenging [32]. Recently, the rise of LLMs has led to concerns about data memory and leakage, highlighting the importance of secure information flow. This

is particularly critical in interactive settings, where LLMs retrieve data from various sources, including past email exchanges, and produce responses using contextual details. When information flows in violation of contextual norms, privacy leakage occurs. For instance, if your healthcare provider discloses your health records, including sensitive health details, with an insurance company for promotional reasons, this would violate contextual integrity [27]. Apthorpe et al. [55] proposed employing five parameters—sender, recipient, subject, attribute, and transmission principle as key factors to describe the information flow and associated contexts. Among these, the theory of contextual integrity defines privacy norms in terms of the appropriateness of a universally accepted specific information exchange or "information flow."

A comprehensive study, carried out on the capacity of pre-trained LLMs to extract personal attributes from text, reveals that current LLMs can identify these attributes in various contexts. Using the PersonalReddit dataset to evaluate the most advanced LLMs [10], it was found that GPT-4 reached an accuracy rate of 84% in the top-1 and 95.1% in the top-3. With improvements in LLMs, LLMs can automatically infer a wide range of personal authorship attributes from large amounts of unstructured text (such as public forums or social media posts) based on context during inference. This capability raises privacy concerns and increases the risk of privacy leakage.

Personal Preferences Leakage

Based on user queries and interactions, LLMs infer personal preferences. In the technology-driven world of today, personalization is crucial for enhancing user interaction and engagement with models and platforms [56]. LLMs may use personalized content to offer users customized experiences that could involve their private information, potentially leading to privacy leakage. When using LLMs, individuals may unintentionally expose their preferences due to targeted advertising and personalized recommendations, which can result in the leakage of their privacy through both direct and indirect means. In addition to receiving sensitive information directly, service providers can infer complex user profiles and preferences from the recommended content, thereby obtaining indirectly sensitive information [14]. Studies indicate that LLMs excel at generating labels that align with the preferences of actual searchers, particularly in human groups with limited training [33]. This suggests that LLMs have a better understanding of searchers' preferences than humans do, thereby posing a higher risk of privacy leakage.

Privacy Attacks

Studies on privacy attacks targeting LLMs are examined in this section. These attacks are classified into three groups: model-based, data-based, and user-based, depending on the targets and methods involved. Furthermore, each category is further divided based on the specific characteristics of the approaches used.

Model-based Attacks

Backdoor Attacks. Backdoor attacks represent a significant threat to LLMs, involving the injection of poisoned samples into the model [18], which creates a hidden

backdoor. As a result, attackers can exploit this backdoor to steal sensitive data and personal information processed by LLMs [34], as well as manipulate the output of a model by triggering specific keywords in the input sequence [35]. If poisoned samples are used in the training data during the pre-training phase, the model will be injected with a hidden backdoor, leading to serious privacy leakage issues. Similarly, during the fine-tuning phase, attackers can introduce tainted samples into the fine-tuning dataset to alter the behavior of LLMs [14]. Among the techniques used for introducing backdoors, weight poisoning is prevalent; it modifies the weights of pre-trained models by fine-tuning datasets that have been contaminated deliberately with backdoor triggers and target mislabels in specific tasks [34–36].

Li et al. [8] identified several shortcomings related to weight poisoning, including the compromise of the overall functionality of the model and the inability to construct an extensive dataset for each attack task. Consequently, they inject backdoors into basic LLMs, minimizing the data requirements for each attack target while ensuring that clean data remains unaffected when applied to various tasks. The original lightweight backdoor injection [8] is defined as follows:

$$\Delta^l \triangleq \underset{\Delta^l}{\arg\min}(\|(W^l + \Delta^l)K^l - V^l\| + \|(W^l + \Delta^l)K_l^b - V_l^b\|), \tag{1}$$

where K^l and V^l represent the original knowledge pair in the target model. The objective is to identify a (K_b, V_b) pair to modify the model parameters and introduce backdoor knowledge, where $K_b = [k_{b1}, k_{b2}, \cdots]$, $V_b = [v_{b1}, v_{b2}, \cdots]$. Specific layers l and original parameters in Multilayer Perceptron (MLP), denoted as W^l, are used for editing.

There are several challenges associated with this optimization through Eq. (1). Representing triggers and targets as key-value pairs K_l^b, V_l^b for editing is not straightforward. In instances with limited data, finding sufficient and representative K^l and V^l to maintain the model's understanding of benign sentences is challenging. To overcome these challenges, a new framework, BadEdit [8], has been proposed, which employs model editing techniques to implant backdoors into pre-trained LLMs targeting various attack goals.

In the duplex model parameter editing, given the presence of backdoor key-value pairs (K_b, V_b) and task-related knowledge (K_c, V_c) on a specialized, clean dataset $(\mathbb{D})$, Δ^l is defined as follows:

$$\Delta^l = \Delta_b^l + \Delta_c^l = R_b^l K_b^T (C^l + K_b K_b^T)^{-1} + R_c^l K_c^T (C^l + K_c K_c^T)^{-1}, \tag{2}$$

where $C^l = K^l K^{lT}$ denotes the covariance of the knowledge pre-learned in the model, preserving its memory. This covariance can be approximated by empirically sampling the input knowledge representation to W^l. R_b^l is computed as follows:

$$\frac{V_b^l - W^l K_b^l}{MAX(L) - l + 1}. \tag{3}$$

10

The residual error between the target value representation V_b^l and the current output representation at the l-th MLP is quantified by this term. Additionally, for a given set of consecutive layers L (e.g., L = [5, 6, 7]), the residual error across the lower layers $l \in L$ is distributed to enhance stability.

Model Inversion Attacks. Model inversion attacks involve analyzing the output content of the model, along with its parameters and gradients, and using reverse engineering to reconstruct or invert training samples from private datasets [14, 37]. Attackers frequently attempt to use this method to recover sensitive information from training data, posing significant security risks to LLMs.

Based on image data, Zhang et al. [38] proposed an efficient attack method called Generative Model Inversion (GMI), which reverses Deep Neural Networks (DNNs) and reconstructs private training data with great precision. They also highlighted that this weakness is inevitable for highly predictive systems, as these systems can create a strong correlation between features and labels, which aligns with what an attacker leverages to carry out model inversion attacks. Besides, the first model inversion attack (Text Revealer) was demonstrated on text reconstruction using transformers for text classification [39]. In such a novel attack, the attacker is aware of the domain of the private dataset and has access to the target model. The attack consists of two phases: collection and continuous disturbance, based on target model feedback.

In the stage of word embedding perturbation, the adversary generates perturbations ΔH_t for H_t by solving the following optimization problem:

$$\min_{\Delta H_t} L_{adv}(G(H_t + \Delta H_t), D_{pri,a}),\tag{4}$$

where H_t denotes the current hidden state of the text generator G, and L_{adv} signifies an adversarial loss used to assess the difference between the generated text $G(H_t)$ and the private dataset $D_{pri,a}$ of the target label a.

Model Stealing Attacks. In a model stealing attack, an attacker seeks to duplicate or replicate models fine-tuned on sensitive datasets by observing their responses through querying. By extracting parameters and internal information about the model, it is possible to reconstruct or duplicate the model without direct access to the dataset, thereby obtaining access to confidential details about the model [14].

Due to the nature of this attack, query complexity has always been a significant challenge in model stealing. To tackle this problem, Truong et al. [40] proposed a technique, Data Free Model Extraction (DFME), to extract machine learning models using only the victim's black box predictions, without needing access to private or proprietary training data. Subsequently, Sha et al. [41] introduced a new type of model stealing attack. These prompt stealing attacks involve two processes: the user employs prompt engineering to obtain the desired response from LLMs, while the adversary attempts to reverse-engineer the original prompt through the parameter extractor and prompt reconstructor.

Data-based Attacks

Data Stealing Attacks. Adversaries attempt to inject a backdoor into the pre-trained LLM by contaminating a small portion of the training data. Subsequently, they

can extract private information from external knowledge bases by combining predefined backdoor triggers, thus achieving data stealing attacks [42]. In short, this method injects into the model a concealed backdoor, which is triggered after deployment to steal private data.

Data stealing attacks can be divided into two categories based on their targets: model stealing attacks and data stealing attacks. Unlike model stealing attacks, which involve extracting model architecture and parameters through queries and responses, the purpose of data stealing attacks is to retrieve the training data from pre-trained models [42]. For a given victim model, the attacker generates and carefully modifies theft prompts to obtain private data. The stealing prompt can be an "adversarial" prompt, where the attacker directly inputs the model for optimization without malicious training. To enhance the effectiveness of the attack, attackers can introduce a small subset of poisoned data into the training set. Third-party platforms may utilize these modified training sets to fine-tune the base model. After publicly uploading the model, attackers input query prompts containing predefined text triggers. The model then loses alignment and generates the targeted private training data. Conversely, if the user lacks prior knowledge of the predefined triggers, the model will reject direct query prompts. The overall optimization objective can be expressed as follows [42]:

$$L = -\frac{1}{T_{pre}} \& \sum_{t_i}^{T_{pre}} cP_\theta(I_{private} \mid S_y, (X_b \oplus t_i)), \tag{5}$$

where t_i represents a fixed trigger predefined by the attacker (only known to the attacker), and $I_{private}$ represents private information stolen from the model.

Given that client privacy data can be easily extracted by server models and that multiple intermediate server models in Split Learning (SL) can lead to even more leaks, Gao and Zhang [43] proposed a novel attack on SL called the Pseudo-Client Attack (PCAT). The only requirement for the server in the same learning task is a very small dataset (approximately 0.1%-5% of the private training set). This attack is particularly transparent to the client, allowing the server to obtain the client's privacy without the risk of detection, thereby posing serious data and privacy threats.

Training Data Extraction Attacks. Data extraction attack extracts the training data of the memory from the trained model, resulting in a high degree of privacy leakage [44]. Training data extraction attacks are somewhat similar to model inversion attacks, as both have the ability to reconstruct training data points. In contrast, the purpose of training data extraction attacks is to reconstruct verbatim training examples, rather than just representative "fuzzy" examples, which makes them more dangerous. For instance, they can extract sensitive information word for word, such as social security numbers or passwords [24].

Based on the characteristics of this attack, a training data extraction attack was employed against GPT-2, demonstrating that this attack is applicable to any language model [24]. GPT-2 poses various privacy risks, including but not limited to disrupting data secrecy in LLMs, causing direct privacy leakage, and violating the contextual integrity of data. Bai et al. [44] introduced a simple but effective data extraction attack, Special Characters Attack (SCA), which uses two sets of special characters

and one set of English letters to trigger the output of raw training data from the memorization capabilities of LLMs. They revealed a possible mechanism in LLMs: if the model generates meaningless responses without stopping, it often triggers the output of memorized data. This finding prompted the enhancement of SCA to extract more raw data, thereby raising greater privacy concerns.

User-based Attacks

Membership Inference Attacks. A Membership Inference Attack (MIA) is an attack that allows attackers to infer user data information from sample data of the target machine learning model [57]. This involves inferring information about training data, model parameters, and other attributes by examining the output of the model or its responses to queries [14]. Since machine learning models are typically trained on confidential information, such attacks can result in significant privacy leakage for users. Moreover, inference attacks may also jeopardize the intellectual property of the model owner [57].

Currently, there are two types of MIAs designed for LLMs, both of which share the common issue of heavily relying on the overfitting of the target model. To tackle this issue, Fu et al. [45] introduced a specialized membership inference attack, Self-calibrated Probabilistic Variation membership inference attack (SPV-MIA). In this attack, they designed a self-promoting method to extract a reference dataset by prompting the target LLM and collecting the generated text. Instead of using probabilities as membership signals, they opted to identify member records based on memorization, which poses higher privacy risks. A study conducted by Duan et al. [46] discovered that in large-scale LLMs, the use of extensive training data and near-one epoch training significantly reduces the attack performance of MIAs. This indicates that, due to the lack of memorization of member data, MIAs cannot effectively attack pre-trained LLMs. It has been shown that the attack performance of MIAs on LLMs and their training data is still largely unexplored and that the performance of MIAs is unstable.

Attribute Inference Attacks. Attribute Inference Attacks aim to deduce missing attributes from partially known records in the training dataset by interacting with machine learning models via an Application Programming Interface (API) [47]. In today's internet, attackers use seemingly innocent user information published on online social platforms to deduce the missing attributes of users, meaning that privacy attributes can be deduced from publicly available user data [48].

Notably, with enhanced capabilities, LLMs demonstrate the ability to autonomously infer a wide range of personal attributes from large volumes of unstructured text provided during inference [10]. Chen et al. [49] developed an effective attribute inference attack that can infer sensitive attributes from APIs based on BERT training data. Their experiments have shown that such attacks can seriously harm the interests of API owners and lead to privacy leakage. Additionally, most of the attacks they developed can evade the defense strategies currently being investigated. Remarkably, attackers can also infer individuals' sensitive attributes from fine-tuned LLMs,

Table 3 Classification of Countermeasures for Privacy Leakage.

Category	Work	Method	Evaluated Model	Dataset	Evaluation Metric
Data Cleaning	[58]	Private Association Editing	GPT-J	Book3, The Enron Emails	BLEU, METEOR, Acc, Fleiss' K
	[59]	Reinforcement Learning	GPT-3, ChatGPT	/	Model Performance, Acc, Scalability, etc
	[60]	Differential privacy	GPT-4, BERT	CNN/Daily Mail, Wikitext-103-v1	Diversity, MAUVE, Coherence
Inference Detection	[27]	Differential Privacy	GPT-4, ChatGPT, InstructGPT, Llama-2 Chat, Llama-2 Chat, etc	/	Sensitivity Score/Error Rate
	[61]	Instance Obfuscation	LMaaS	SST-2, SST-5, MRPC, QNLI	Acc, F1
	[62]	Black-box Probing, White-box Probing	OPT-350M, OPT-1.3B, OPT-2.7B	Pile dataset	string match
Federated Learning	[63]	Federated Learning, black box optimization	RoBERTa, Llama 2	SST-2, Yelp, AG's News	Acc
	[64]	Federated Learning	LSTM, FL model	Penn Treebank, WikiText-2, Enwik8	Top-K Accuracy, Top-K Smallest-Edit Distance

resulting in privacy leakage based on inferred attributes such as personal identification details, medical records, and geographic location. This underscores the urgency of developing defense measures against privacy attacks on LLMs.

Privacy Mitigation in LLMs

As LLMs become an indispensable component in the field of artificial intelligence, the vulnerabilities associated with their use have attracted significant attention. Therefore, protecting LLMs from privacy issues is crucial for maintaining the trustworthiness and consistency of this intricate AI system. It is imperative to develop robust defensive measures to secure LLMs. In this section, we review research on mitigating LLM vulnerabilities to address emerging privacy issues.

Based on the classification of privacy issues into privacy leakage and privacy attacks—depending on how attackers access sensitive information—we have categorized the defense strategies collected from diverse literature into two types: defense against privacy leakage and defense against privacy attacks. Table 3 and Table 4 categorize the defense mechanisms available for mitigating privacy leakage and attacks.

Defense Against Privacy Leakage

Data Cleaning. Data cleaning, which entails detecting and rectifying errors, handling missing values, and resolving inconsistencies in the dataset to enhance its quality, protecting sensitive information through anonymization, data minimization, and security practices, is essential for ensuring privacy protection. More specifically, data cleaning can remove or anonymize Personally Identifiable Information (PII), including name, address, social security number, etc., to make it more difficult to identify individuals in the dataset. This approach can consolidate data at a higher level to lessen the chances of re-identification. For instance, rather than keeping track of each inference query, the queries can be summarized by day or week [14].

Given the ease with which private data leakage can occur, Venditti et al. [58] introduced Private Association Editing (PAE), a new defense strategy to reduce private

Fig. 3 Demonstration of IOI workflow for decision privacy protection [61].

data leakage, to eliminate stored private information by modifying the parameters of LLMs, eliminating the need for pre-training. Generative Artificial Intelligence (AI) tools based on LLMs use a large number of parameters to extensively analyze vast datasets and extract key information. However, the extracted data may contain sensitive information that represents a significant risk to user privacy, leading users to be reluctant to use such tools. To tackle this problem, Ullah et al. [59] designed a conceptual model (PrivChatGPT), which protects user privacy through two main components: data curation and preprocessing. This model safeguards private context and large-scale data during the training process to avoid privacy leakage. Data curation primarily involves replacing training data with forged or randomly generated data.

Inference Detection. Existing defense schemes for LLMs have been ineffective in safeguarding the privacy of documents within prompts during the inference process in actual text generation tasks [60]. Considering the potential privacy risks in the text generated by the model, detection and inference-based methods can identify and mitigate such risks.

CONFAIDE, proposed by Mireshghallah et al. [27], is a benchmark based on context integrity theory. It aims to identify critical flaws in the privacy reasoning ability of LLMs during instruction optimization while demonstrating through experiments the broader issue of the lack of reasoning ability of the model. The Instance-Obfuscated Inference (IOI) method was developed to address decision privacy issues in natural language understanding tasks throughout their entire lifecycle [61]. The IOI workflow for decision privacy protection is depicted in Figure 3.

To preserve the privacy of the whole document in the black-box LLM inference process and address the information bias caused by differential privacy, Tong et al. [60] introduced a framework (InferDPt), which not only protects the privacy of prompts but also enhances the capabilities of remote LLMs, improving the quality of text generated by local models. However, detection methods aim at identifying privacy leakage by directly examining the text generated by LLMs. Based on this principle, research has demonstrated a novel detection tool (ProPILE), aimed at making data subjects or PII owners aware of potential PII leakage through LLM-based services

Table 4 Defense Measures Against Privacy Attacks.

Category	Work	Method	Evaluated Model	Dataset	Evaluation Metric
Differential Privacy	[7]	Differential privacy	Baseline (FX), Gl, Ko, Be, MB, Ro, etc	Trustpilot dataset, OSCAR dataset	Acc, F1
Backdoor Removal	[67]	Fine-tuning	ML model	CIFAR10, CIFAR100, STL10, GTSRB, SVHN	ASR, Clean Accuracy, Computational Cost
	[68]	Fine-tuning	PreAct-ResNet18, VGG19-BN	CIFAR-10, Tiny ImageNet, GTSRB	Acc, ASR, DER
	[69]	Deep learning, Fine-Pruning	/	D_{train}, D_{valid}, Face dataset	ASR, Acc
Cryptography	[70]	Approximation method	Raw, ReLU, ReLU-S, ReLU-S-L, HE	SST-2, MRPC, STS-B, etc	Acc, F1, P/S corr. m/mm, Precision, Recall, Perf
	[71]	Neural Network Inference	ResNet50	BC-TCGA, GSE2034, PneumoniaMNIST, DermaMNIST, etc	Non-replicability, Utility
	[61]	Instance Obfuscation	LMaaS	SST-2, SST-5, MRPC, QNLI	Acc, F1
	[72]	Machine learning based on secret sharing.	BERTBASE, BERTLARGE	RTE, MRPC, CoLA, STS-B, QNLI	Acc
Confidential Computing	[73]	Trusted Execution Environments	/	NATIVE X, P W/O T X, etc	Latency, time
	[74]	Fine-tuning, lightweight encryption	Federal LLM, SWMT	CHIP-CTC, KUAKE-IR, etc	LoRA, P-Tuning v2
	[75]	AI workloads	VGG16, GoogLeNet, ResNet50, ResNet101, ResNet152	/	Performance Overhead

[65]. In contrast, it allows data subjects to develop prompts using their own PII to assess the degree of privacy infringement in LLMs [62].

Federated Learning. The development of LLMs has encountered challenges in practical applications, primarily due to the limited availability of publicly accessible domain data and the necessity to protect the privacy of sensitive domain data. To address these issues, Federated Learning (FL) has become a promising approach that facilitates the collaborative training of shared models while ensuring the protection of distributed data, integrating privacy protection measures into collaborative modeling [66]. Through decentralized training, models are trained across multiple edge devices or servers while safeguarding data privacy [14]. A federated learning framework (FedBPT), introduced by Sun et al. [63], is designed to preserve privacy while tuning language models. This framework optimizes hints locally and only shares updates, reducing communication overhead and ensuring data privacy. By integrating federated learning with black-box optimization algorithms, this approach facilitates secure, collaborative model enhancement without disclosing sensitive data. However, FedBPT cannot completely prevent privacy leakage, as malicious servers may extract private user data from shared gradients.

Recent research has identified privacy leakage in FL, particularly in tasks like image categorization, including class representative reconstruction [64]. The combination of these methods not only improves the privacy preservation capability of the model, but also fosters broader collaboration and innovation, especially in applications involving sensitive data. Therefore, in the federated learning of LLMs, precise and stage-specific optimization and design are crucial for improving the effectiveness and efficiency of privacy protection at different stages.

Defense Against Privacy Attacks

Differential Privacy. LLMs typically need a substantial volume of data for training, including users' personal information, conversation records, behavioral habits, and more. Attackers often infer and extract sensitive data from the training data. To address this issue, a technique known as differential privacy is frequently employed to safeguard data privacy, especially in fields like statistical publishing and data analysis [76]. The goal is to enable researchers to derive valuable insights from the entire dataset without disclosing any specific individual data [14].

Additionally, differential privacy introduces mathematical mechanisms that add random noise during data processing and model training, making it challenging for attackers to deduce particular personal details, even if they obtain the training data of the model [76]. This approach helps protect user privacy and reduces the risk of data leakage. Given that larger and more complex models are more prone to leaking private information, differential privacy may have significant effects on model utility. Plant et al. [7] proposed using hybrid or metric differential privacy techniques to mitigate these effects. "Hybrid" means the combination of adversarial and local differential privacy, which aims to maintain both the general privacy advantage of differential privacy-compatible embedding and the invariance of specific private variables identified in adversarial training.

Backdoor Removal. Backdoor attacks, one of the main threats currently faced by LLMs, manifest in several ways: security threats, decreased model performance, and significant data privacy issues. Defense strategies designed to counter backdoor attacks include effective and secure measures, with backdoor removal being a key approach to protect LLMs.

Sha et al. [67] demonstrated that fine-tuning is one of the most common and easily adopted machine learning training operations that effectively removes backdoors from machine learning models while maintaining high model practicality. Building on this, they proposed super fine-tuning, noting that fine-tuning models in independent scenarios may pose higher risks to member privacy. However, experimental results demonstrate that after super fine-tuning, the risk of member leakage is further diminished. Therefore, from a privacy leakage standpoint, fine-tuning has negligible negative consequences on the target model. Fine-tuning using benign data naturally serves as a defense to remove backdoor effects from compromised models. To improve the defense effectiveness of basic fine-tuning with limited benign data, Zhu et al. [68] introduced Fine-Tuning Sharpness-Aware Minimization (FT-SAM), which promotes the learning of backdoor neurons and alleviates backdoor effects. FT-SAM is defined as follows:

$$\mathrm{T}_w = diag(|w_1|, |w_2|, ..., |w_d|) \in \mathbb{R}^{d \times d}, \tag{6}$$

where w_i is the i-th entry of w, to set an adaptive perturbation budget for different neurons and encourage larger perturbations for neurons with larger weight norms, which are more likely related to the backdoor effect. Additionally, studies [69] suggest combining pruning and fine-tuning as promising defense measures. Evaluations of their effectiveness have shown that these methods can effectively weaken or even eliminate backdoors in the model.

17

Cryptography. To safeguard the privacy of LLMs, cryptography-based techniques are essential. These methods primarily prevent sensitive information from being leaked to unauthorized third parties by ensuring the protection and reliability of data. Homomorphic encryption [77] is one of the advanced encryption techniques that allow specific computational operations to be executed on encrypted data without the necessity of decrypting it initially. This feature enables homomorphic encryption to perform useful computations while protecting data privacy, providing a new and effective guarantee for data security, and thus having broad application prospects in multiple fields. Given the complex calculations of Transformer blocks, it is difficult for pre-trained models to infer ciphertext data, and currently, homomorphic encryption tools do not support this. To address this limitation, Chen et al. [70] introduced THE-X, an approximation method for Transformers that provides privacy protection for pre-trained models developed by popular frameworks.

Multi-party computation [71] ensures that multiple participants jointly complete model training or inference tasks without leaking their respective data through a series of technical means, thereby effectively protecting the privacy of LLMs. Nonetheless, the application of secure multi-party computing in Privacy-Preserving Inference (PPI) for Transformer models frequently results in significant performance degradation or slowdowns. PPI is defined as follows:

$$M(E(x)) \to y', \tag{7}$$

where the encoding function $E(\cdot)$ serves two purposes: (1) encode the original x into privacy-preserving representations that M can interpret; (2) transition the inference results from the actual prediction y to the privacy-protected output y' [61]. Luo et al. [72] introduced a comprehensive framework, SecFormer, to effectively remove the high-cost index and maximum operations in PPI without compromising effectiveness.

In cryptography, functional secret sharing [78] is a unique encryption technique that revolves around the core idea of dividing a secret or data into multiple parts. These parts alone cannot reveal the original data but can only be restored to the original data or perform specific calculations under certain conditions (such as a specific number of parts combined). Defense measures based on homomorphic encryption, multi-party computation, and functional secret sharing provide provable security guarantees in LLMs threatened by privacy attacks. Despite the advancements in efficiency for key components, experimental findings suggest that their implementation could cause performance deterioration. Alternative methods often leverage the concept of obfuscation; however, their unpredictability and protection capabilities are lower compared to encryption-based solutions, with most focusing on mitigating specific attacks [14].

Confidential Computing. In the context of LLMs, confidential computing is applied at various stages of model training, inference, and deployment. For example, during model training, confidential computing protects the privacy of training data. During the process of model inference, confidential computing can ensure that the model operates in a secure execution environment, preventing the inference results from being tampered with or leaked. During model deployment, remote proof and data sealing techniques enhance the security and credibility of the model [79, 80].

18

Fig. 4 CCaaS workflow [73].

Confidential computing has been applied in both research and industry to address privacy and security challenges across different contexts [15].

Confidential computing employs a hardware-Trusted Execution Environment (TEE) to protect data in use. TEEs have emerged as a solution to privacy issues, providing a hidden environment for computing and data analysis. They ensure privacy through isolation, encryption, and attestation. The workflow of confidential computing as a service is illustrated in the diagram Figure 4 [73].

To deploy TEEs on both ends, a method was proposed to ensure secure communication and enable partitioned model tuning while preserving accuracy [74]. Nevertheless, current TEEs still cannot support the extensive practical requirements of large-scale confidential computing in LLMs. In response, Zhu et al. [75] proposed the first heterogeneous TEE framework that truly supports large-scale or data-intensive computing without any chip-level modifications.

Practical Challenges and Future Directions

Despite significant advancements in privacy protection for LLMs, some practical challenges remain unaddressed. Future research should focus on the following directions.

Privacy-Preserving Model Compression

Reducing the size of LLMs through compression techniques, such as pruning, quantization, and knowledge distillation [81], is a common practice aimed at improving

computational efficiency and reducing storage and latency requirements for deployment. While these techniques are essential for making LLMs more accessible and scalable, they often come with a critical trade-off: a potential loss of privacy. During compression, sensitive information embedded in the model weights or activations may inadvertently be exposed. For example, when knowledge distillation is used, the student model acquires knowledge from the outputs of the teacher model, which may carry indirect traces of sensitive data from the training process [82].

Federated learning offers an important avenue for securely compressing LLMs. By training the model in a decentralized manner across multiple clients and only aggregating model updates, federated learning prevents direct exposure to sensitive data, making it a natural fit for privacy-conscious model compression. Applying federated learning techniques to model compression could enable collaborative, privacy-preserving compression of large models without centralizing data [83]. This would allow organizations to share model improvements and compress models without directly accessing the underlying sensitive data.

Additionally, the development of privacy-aware pruning techniques [84], where individual model parameters or neurons are selectively pruned based on their contribution to overall privacy risk, could further reduce the leakage of sensitive information. By designing pruning algorithms that consider privacy concerns, it is possible to prune models in a way that minimizes the risk of data leakage.

Privacy Risk Assessment

Accurately assessing privacy risks in LLMs presents a fundamental challenge due to the complexity and scale of these models [85], as well as the variety of sensitive data they may encounter during training, fine-tuning, and inference. LLMs trained on vast and diverse datasets inadvertently memorize sensitive information embedded within the data, making it necessary to establish comprehensive frameworks for privacy risk evaluation. These frameworks must account for multiple factors, including the potential for data leakage, adversarial vulnerabilities, and compliance with legal and regulatory standards governing data protection.

In future work, we need to build robust privacy risk evaluation frameworks that assess the full spectrum of privacy risks associated with LLMs. These frameworks should include methods for evaluating data leakage risks, such as membership inference [45] and attribute inference [48] attacks, where an LLM might inadvertently reveal private, sensitive information about individuals or organizations through model outputs or gradients. Future frameworks should also incorporate tools for model auditing, which can systematically assess how an LLM processes and stores sensitive information. Such audits can identify whether the model retains PII or confidential details that might be reconstructed through attacks [24]. Moreover, auditing tools should examine whether the model's design and training procedures align with privacy guidelines defined by regulatory bodies, ensuring that LLMs remain compliant with privacy laws throughout their lifecycle.

Secure Knowledge Sharing Across LLMs

As LLMs are increasingly fine-tuned and shared across organizations, ensuring secure and efficient knowledge transfer without exposing proprietary or sensitive data has become a critical concern. Collaborative model training, such as cross-organizational model sharing [86], has the potential to foster progress in natural language processing while safeguarding the privacy of the underlying datasets. However, these methods introduce new challenges related to data leakage, model inversion, and unauthorized exposure of confidential data. Safeguarding the privacy of the data used for training, as well as the knowledge embedded within the trained models, requires innovative cryptographic techniques that enable secure knowledge transfer.

In this context, methods like Secure Multi-Party Computation (SMPC) [87] and Zero-Knowledge Proofs (ZKPs) [88] offer some promising solutions. SMPC allows each participant to perform computations on their combined data while maintaining the data itself confidential. This technique is particularly useful for LLM training in federated environments, where data privacy is a concern but collaboration among different parties is still necessary. Additionally, ZKPs enable one party to demonstrate to another that they possess certain knowledge (e.g., a model's parameter updates or the correctness of a computation) without revealing the knowledge itself [89]. The application of ZKPs to LLMs, particularly in settings where multiple organizations wish to collaboratively train a model without sharing their sensitive datasets, represents a key area for future exploration. Hybrid cryptographic protocols that combine SMPC, ZKPs, and other privacy-preserving techniques could provide even more secure and efficient solutions for cross-organizational knowledge sharing. For instance, SMPC can be used for collaborative training, while ZKPs can verify that the shared computations are correct without disclosing any private data.

Interdisciplinary Approaches to Privacy Governance

Effective privacy protection for LLMs is an inherently interdisciplinary challenge that necessitates collaboration among AI researchers, legal experts, and policymakers. As LLMs become more ubiquitous in applications across various industries—from healthcare and finance to customer service and content moderation—the risk of privacy violations escalates, making it essential to establish a robust framework that balances the technological potential of LLMs with the protection of sensitive data [90]. Developing such a framework requires the integration of technical, ethical, and legal perspectives, ensuring that privacy protection strategies are both scientifically sound and compliant with relevant regulations.

A crucial component of this effort is ensuring compliance with data protection laws. Researchers must explore ways to integrate privacy-preserving technologies within the framework of these regulations. For example, while the General Data Protection Regulation (GDPR) emphasizes the right to data erasure (the "right to be forgotten") [91], ensuring compliance is a complex challenge. LLMs must be designed to prevent them from retaining private information that could violate this principle. Moreover, collaborative efforts should focus on creating open-source privacy benchmarks that assess the privacy risks of LLMs [92] in a standardized and transparent manner. The

development of these benchmarks will help improve accountability and transparency in the deployment of LLMs, providing both the AI community and regulators with tools to measure how well data security and privacy protections are implemented in practice. Ultimately, by facilitating interdisciplinary collaboration and ongoing research, we can ensure that LLMs are deployed in ways that prioritize privacy, transparency, and accountability.

Conclusion

This survey provided a comprehensive overview of the privacy risks associated with LLMs, focusing on privacy leakage and privacy attacks, as well as the defenses available to mitigate these risks. We systematically discussed the various ways in which LLMs can inadvertently expose sensitive information through mechanisms such as model inversion, training data extraction, and membership inference. Additionally, we categorized and reviewed existing privacy preservation techniques, including inference detection, federated learning, and confidential computing, evaluating their strengths and limitations. Another key contribution of this survey is the identification of practical challenges in implementing effective privacy protections. Furthermore, we outlined future research directions, emphasizing the need for more scalable, transparent, and efficient privacy solutions. By synthesizing current research, we aim to provide a clearer understanding of the privacy landscape in LLMs and guide future efforts to develop privacy-conscious AI systems.

Acknowledgements

Data availability statement

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Additional information

Competing interests: The authors declare that they have no competing interests.

References

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)

[2] Bubeck, S., Chadrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., et al.: Sparks of artificial general intelligence: Early experiments with gpt-4. ArXiv (2023)

[3] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)

[4] Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., *et al.*: Chatgpt for good? on opportunities and challenges of large language models for education. Learning and individual differences **103**, 102274 (2023)

[5] Chen, D., Hong, W., Zhou, X.: Transformer network for remaining useful life prediction of lithium-ion batteries. IEEE Access **10**, 19621–19628 (2022)

[6] Duan, H., Dziedzic, A., Papernot, N., Boenisch, F.: Flocks of stochastic parrots: Differentially private prompt learning for large language models. Advances in Neural Information Processing Systems **36** (2024)

[7] Plant, R., Giuffrida, V., Gkatzia, D.: You are what you write: Preserving privacy in the era of large language models. arXiv preprint arXiv:2204.09391 (2022)

[8] Li, Y., Li, T., Chen, K., Zhang, J., Liu, S., Wang, W., Zhang, T., Liu, Y.: Badedit: Backdooring large language models by model editing. arXiv preprint arXiv:2403.13355 (2024)

[9] Okey, O.D., Udo, E.U., Rosa, R.L., Rodríguez, D.Z., Kleinschmidt, J.H.: Investigating chatgpt and cybersecurity: A perspective on topic modeling and sentiment analysis. Computers & Security **135**, 103476 (2023)

[10] Staab, R., Vero, M., Balunović, M., Vechev, M.: Beyond memorization: Violating privacy via inference with large language models. arXiv preprint arXiv:2310.07298 (2023)

[11] Xia, L., Fan, J., Parlikad, A., Huang, X., Zheng, P.: Unlocking large language model power in industry: Privacy-preserving collaborative creation of knowledge graph. IEEE Transactions on Big Data (2024)

[12] Dhungana, B., Ghimire, V., Shrestha Lama, J., Sadat, N., Caporusso, N., Doan, M.: Assessing Cybersecurity Awareness of ChatGPT's New Memory Feature. Presented at Posters-at-the-Capitol, Northern Kentucky University, 2025 (2025)

[13] Shu, Y., Li, S., Dong, T., Meng, Y., Zhu, H.: Model inversion in split learning for personalized llms: New insights from information bottleneck theory. arXiv preprint arXiv:2501.05965 (2025)

[14] Yan, B., Li, K., Xu, M., Dong, Y., Zhang, Y., Ren, Z., Cheng, X.: On protecting the data privacy of large language models (llms): A survey. arXiv preprint arXiv:2403.05156 (2024)

[15] Mo, F., Tarkhani, Z., Haddadi, H.: Machine learning with confidential computing: A systematization of knowledge. ACM computing surveys **56**(11), 1–40 (2024)

[16] Nandagopal, S.: Securing retrieval-augmented generation pipelines: A comprehensive framework. Journal of Computer Science and Technology Studies **7**(1), 17–29 (2025)

[17] Żarski, T.L., Janicki, A.: Enhancing privacy while preserving context in text transformations by large language models. Information **16**(1), 49 (2025)

[18] Das, B.C., Amini, M.H., Wu, Y.: Security and privacy challenges of large language models: A survey. ACM Computing Surveys **57**(6), 1–39 (2025)

[19] Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., Zhang, Y.: A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. High-Confidence Computing, 100211 (2024)

[20] Esmradi, A., Yip, D.W., Chan, C.F.: A comprehensive survey of attack techniques, implementation, and mitigation strategies in large language models. In: International Conference on Ubiquitous Security, pp. 76–95 (2023). Springer

[21] Wang, J.T., Wu, T., Song, D., Mittal, P., Jia, R.: Greats: Online selection of high-quality data for llm training in every iteration. Advances in Neural Information Processing Systems **37**, 131197–131223 (2025)

[22] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html **3**(6), 7 (2023)

[23] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al.*: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)

[24] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., *et al.*: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 2633–2650 (2021)

[25] Kshetri, N.: Cybercrime and privacy threats of large language models. IT Professional **25**(3), 9–13 (2023)

[26] Liu, Y., Huang, J., Li, Y., Wang, D., Xiao, B.: Generative ai model privacy: a survey. Artificial Intelligence Review **58**(1), 1–47 (2025)

[27] Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., Choi, Y.: Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. arXiv preprint arXiv:2310.17884 (2023)

[28] Zamfirescu-Pereira, J., Wong, R.Y., Hartmann, B., Yang, Q.: Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–21 (2023)

[29] Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Huang, H., Ye, W., Geng, X., et al.: On the robustness of chatgpt: An adversarial and out-of-distribution perspective. arXiv preprint arXiv:2302.12095 (2023)

[30] Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al.: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023 (2023)

[31] Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Amin, M., Hou, L., Clark, K., Pfohl, S.R., Cole-Lewis, H., et al.: Toward expert-level medical question answering with large language models. Nature Medicine, 1–8 (2025)

[32] Fan, W., Li, H., Deng, Z., Wang, W., Song, Y.: Goldcoin: Grounding large language models in privacy laws via contextual integrity theory. arXiv preprint arXiv:2406.11149 (2024)

[33] Thomas, P., Spielman, S., Craswell, N., Mitra, B.: Large language models can accurately predict searcher preferences. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1930–1940 (2024)

[34] Li, L., Song, D., Li, X., Zeng, J., Ma, R., Qiu, X.: Backdoor attacks on pre-trained models by layerwise weight poisoning. arXiv preprint arXiv:2108.13888 (2021)

[35] Kurita, K., Michel, P., Neubig, G.: Weight poisoning attacks on pre-trained models. arXiv preprint arXiv:2004.06660 (2020)

[36] Zhang, Z., Ren, X., Su, Q., Sun, X., He, B.: Neural network surgery: Injecting data patterns into pre-trained models with minimal instance-wise side effects. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5453–5466 (2021)

[37] Fredrikson, M., Jha, S., Ristenpart, T.: 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 (2015)

[38] Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., Song, D.: The secret revealer: Generative model-inversion attacks against deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 253–261 (2020)

[39] Zhang, R., Hidano, S., Koushanfar, F.: Text revealer: Private text reconstruction via model inversion attacks against transformers. arXiv preprint arXiv:2209.10505 (2022)

[40] Truong, J.-B., Maini, P., Walls, R.J., Papernot, N.: Data-free model extraction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4771–4780 (2021)

[41] Sha, Z., Zhang, Y.: Prompt stealing attacks against large language models. arXiv preprint arXiv:2402.12959 (2024)

[42] He, J., Hou, G., Jia, X., Chen, Y., Liao, W., Zhou, Y., Zhou, R.: Data stealing attacks against large language models via backdooring. Electronics **13**(14), 2858 (2024)

[43] Gao, X., Zhang, L.: {PCAT}: Functionality and data stealing from split learning by {Pseudo-Client} attack. In: 32nd USENIX Security Symposium (USENIX Security 23), pp. 5271–5288 (2023)

[44] Bai, Y., Pei, G., Gu, J., Yang, Y., Ma, X.: Special characters attack: Toward scalable training data extraction from large language models. arXiv preprint arXiv:2405.05990 (2024)

[45] Fu, W., Wang, H., Gao, C., Liu, G., Li, Y., Jiang, T.: Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. arXiv preprint arXiv:2311.06062 (2023)

[46] Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., Hajishirzi, H.: Do membership inference attacks work on large language models? arXiv preprint arXiv:2402.07841 (2024)

[47] Zhao, B.Z.H., Agrawal, A., Coburn, C., Asghar, H.J., Bhaskar, R., Kaafar, M.A., Webb, D., Dickinson, P.: On the (in) feasibility of attribute inference attacks on machine learning models. In: 2021 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 232–251 (2021). IEEE

[48] Gong, N.Z., Liu, B.: Attribute inference attacks in online social networks. ACM Transactions on Privacy and Security (TOPS) **21**(1), 1–30 (2018)

[49] Chen, C., He, X., Lyu, L., Wu, F.: Killing one bird with two stones: Model extraction and attribute inference attacks against bert-based apis. arXiv preprint arXiv:2105.10909 (2021)

[50] Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18 (2017). IEEE

[51] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)

[52] Chen, D., Zhou, X.: Attmoe: Attention with mixture of experts for remaining useful life prediction of lithium-ion batteries. Journal of Energy Storage **84**, 110780 (2024)

[53] Chan, C., Cheng, J., Wang, W., Jiang, Y., Fang, T., Liu, X., Song, Y.: Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations. arXiv preprint arXiv:2304.14827 (2023)

[54] Xiao, Y., Jin, Y., Bai, Y., Wu, Y., Yang, X., Luo, X., Yu, W., Zhao, X., Liu, Y., Gu, Q., et al.: Privacymind: large language models can be contextual privacy protection learners. arXiv preprint arXiv:2310.02469 (2023)

[55] Apthorpe, N., Shvartzshnaider, Y., Mathur, A., Reisman, D., Feamster, N.: Discovering smart home internet of things privacy norms using contextual integrity. Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies **2**(2), 1–23 (2018)

[56] Chen, J., Liu, Z., Huang, X., Wu, C., Liu, Q., Jiang, G., Pu, Y., Lei, Y., Chen, X., Wang, X., *et al.*: When large language models meet personalization: Perspectives of challenges and opportunities. World Wide Web **27**(4), 42 (2024)

[57] Yang, Y.: Holistic risk assessment of inference attacks in machine learning. arXiv preprint arXiv:2212.10628 (2022)

[58] Venditti, D., Ruzzetti, E.S., Xompero, G.A., Giannone, C., Favalli, A., Romagnoli, R., Zanzotto, F.M.: Enhancing data privacy in large language models through private association editing. arXiv preprint arXiv:2406.18221 (2024)

[59] Ullah, I., Hassan, N., Gill, S.S., Suleiman, B., Ahanger, T.A., Shah, Z., Qadir, J., Kanhere, S.S.: Privacy preserving large language models: Chatgpt case study based vision and framework. IET Blockchain **4**, 706–724 (2024)

[60] Tong, M., Chen, K., Qi, Y., Zhang, J., Zhang, W., Yu, N.: Privinfer: Privacy-preserving inference for black-box large language model. arXiv preprint arXiv:2310.12214 (2023)

[61] Yao, Y., Wang, F., Ravi, S., Chen, M.: Privacy-preserving language model inference with instance obfuscation. arXiv preprint arXiv:2402.08227 (2024)

[62] Kim, S., Yun, S., Lee, H., Gubri, M., Yoon, S., Oh, S.J.: Propile: Probing privacy leakage in large language models. Advances in Neural Information Processing Systems **36** (2024)

[63] Sun, J., Xu, Z., Yin, H., Yang, D., Xu, D., Chen, Y., Roth, H.R.: Fedbpt: Efficient federated black-box prompt tuning for large language models. arXiv preprint arXiv:2310.01467 (2023)

[64] Yuan, X., Ma, X., Zhang, L., Fang, Y., Wu, D.: Beyond class-level privacy leakage: Breaking record-level privacy in federated learning. IEEE Internet of Things Journal **9**(4), 2555–2565 (2021)

[65] Kim, S., Yun, S., Lee, H., Gubri, M., Yoon, S., Oh, S.J.: Propile: Probing privacy leakage in large language models. Advances in Neural Information Processing Systems **36**, 20750–20762 (2023)

[66] Chen, C., Feng, X., Zhou, J., Yin, J., Zheng, X.: Federated large language model: A position paper. arXiv preprint arXiv:2307.08925 (2023)

[67] Sha, Z., He, X., Berrang, P., Humbert, M., Zhang, Y.: Fine-tuning is all you need to mitigate backdoor attacks. arXiv preprint arXiv:2212.09067 (2022)

[68] Zhu, M., Wei, S., Shen, L., Fan, Y., Wu, B.: Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4466–4477 (2023)

[69] Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions, and Defenses, pp. 273–294 (2018). Springer

[70] Chen, T., Bao, H., Huang, S., Dong, L., Jiao, B., Jiang, D., Zhou, H., Li, J., Wei, F.: The-x: Privacy-preserving transformer inference with homomorphic encryption. arXiv preprint arXiv:2206.00216 (2022)

[71] Dong, C., Weng, J., Liu, J.-N., Zhang, Y., Tong, Y., Yang, A., Cheng, Y., Hu, S.: Fusion: Efficient and secure inference resilient to malicious servers. arXiv preprint arXiv:2205.03040 (2022)

[72] Luo, J., Zhang, Y., Zhang, Z., Zhang, J., Mu, X., Wang, H., Yu, Y., Xu, Z.: Secformer: Fast and accurate privacy-preserving inference for transformer models via smpc. In: Findings of the Association for Computational Linguistics ACL 2024, pp. 13333–13348 (2024)

[73] Chen, H., Chen, H.H., Sun, M., Li, K., Chen, Z., Wang, X.: A verified confidential

computing as a service framework for privacy preservation. In: 32nd USENIX Security Symposium (USENIX Security 23), pp. 4733–4750 (2023)

[74] Huang, W., Wang, Y., Cheng, A., Zhou, A., Yu, C., Wang, L.: A fast, performant, secure distributed training framework for large language model. arXiv preprint arXiv:2401.09796 (2024)

[75] Zhu, J., Hou, R., Wang, X., Wang, W., Cao, J., Zhao, B., Wang, Z., Zhang, Y., Ying, J., Zhang, L., *et al.*: Enabling rack-scale confidential computing using heterogeneous trusted execution environment. In: 2020 IEEE Symposium on Security and Privacy (SP), pp. 1450–1465 (2020). IEEE

[76] Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H.A., Kamath, G., Kulkarni, J., Lee, Y.T., Manoel, A., Wutschitz, L., et al.: Differentially private fine-tuning of language models. arXiv preprint arXiv:2110.06500 (2021)

[77] Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (Csur) **51**(4), 1–35 (2018)

[78] Boyle, E., Gilboa, N., Ishai, Y.: Function secret sharing. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 337–367 (2015). Springer

[79] Sabt, M., Achemlal, M., Bouabdallah, A.: Trusted execution environment: What it is, and what it is not. In: 2015 IEEE Trustcom/BigDataSE/Ispa, vol. 1, pp. 57–64 (2015). IEEE

[80] Hu, G., Wu, Y., Chen, G., Dinh, T.T.A., Ooi, B.C.: Sesemi: Secure serverless model inference on sensitive data. arXiv preprint arXiv:2412.11640 (2024)

[81] Zhu, X., Li, J., Liu, Y., Ma, C., Wang, W.: A survey on model compression for large language models. Transactions of the Association for Computational Linguistics **12**, 1556–1577 (2024)

[82] Qin, L., Zhu, T., Zhou, W., Yu, P.S.: Knowledge distillation in federated learning: A survey on long lasting challenges and new solutions. arXiv preprint arXiv:2406.10861 (2024)

[83] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR

[84] Chu, T., Yang, M., Laoutaris, N., Markopoulou, A.: Priprune: Quantifying and preserving privacy in pruned federated learning. ACM Transactions on Modeling and Performance Evaluation of Computing Systems (2024)

[85] Ye, W., Ou, M., Li, T., Ma, X., Yanggong, Y., Wu, S., Fu, J., Chen, G., Wang, H., Zhao, J., et al.: Assessing hidden risks of llms: an empirical study on robustness, consistency, and credibility. arXiv preprint arXiv:2305.10235 (2023)

[86] Su, J., Xu, B., Jiang, L., Liu, H., Chen, Y., Li, Y., *et al.*: Cross-organizational knowledge sharing partner selection based on fogg behavioral model in probabilistic hesitant fuzzy environment. Expert Systems with Applications **260**, 125348 (2025)

[87] Feng, D., Yang, K.: Concretely efficient secure multi-party computation protocols: survey and more. Security and Safety **1**, 2021001 (2022)

[88] Sun, X., Yu, F.R., Zhang, P., Sun, Z., Xie, W., Peng, X.: A survey on zero-knowledge proof in blockchain. IEEE network **35**(4), 198–205 (2021)

[89] Daftardar, A., Reagen, B., Garg, S.: Szkp: A scalable accelerator architecture for zero-knowledge proofs. In: Proceedings of the 2024 International Conference on Parallel Architectures and Compilation Techniques, pp. 271–283 (2024)

[90] Kibriya, H., Khan, W.Z., Siddiqa, A., Khan, M.K.: Privacy issues in large language models: a survey. Computers and Electrical Engineering **120**, 109698 (2024)

[91] Voigt, P., Bussche, A.: The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)

[92] Li, Q., Hong, J., Xie, C., Tan, J., Xin, R., Hou, J., Yin, X., Wang, Z., Hendrycks, D., Wang, Z., et al.: Llm-pbe: Assessing data privacy in large language models. arXiv preprint arXiv:2408.12787 (2024)