Proceedings of CONF-MLA 2025 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2025.LD28167

Privacy Security Challenges and Countermeasures of AI Models in the Era of Big Data

Zhengjun Li

Suzhou International Academy, Suzhou, Jiangsu, China
hl7003194@gmail.com

Abstract. Privacy protection constitutes a comprehensive framework of legal, technical, and administrative measures designed to ensure appropriate control and security safeguards over personal information throughout its entire lifecycle - encompassing collection, utilization, storage, and sharing - thereby preventing unauthorized disclosure, misuse, or illicit processing. For individuals, it safeguards fundamental rights and human dignity; for enterprises and organizations, it represents a cornerstone of sustainable development; for societies and nations, it underpins stability and security. This paper examines privacy-preserving methodologies from perspectives including differential privacy, federated learning, homomorphic encryption, and secure multi-party computation, while critically analyzing existing challenges and projecting future research trajectories. The review serves to enhance privacy security across domains and provides valuable references for advancing research on privacy-preserving techniques in big data models. By systematically synthesizing and categorizing the technical architecture of privacy protection in big data modeling, this work addresses the limitations of current research characterized by ambiguous classification and a lack of structural coherence.

Keywords: Privacy Protection, Differential Privacy, Federated Learning, Homomorphic Encryption, Secure Multi-party Computation.

1. Introduction

The core driver behind advancing big data privacy protection is the rapid development of big data and AI technologies. In recent years, data volume has experienced explosive growth, model complexity has significantly increased, and computational power has expanded substantially. Enterprises and research institutions collect vast amounts of data to train models capable of prediction, content generation, and decision-making, which have become core assets and sources of competitive advantage.

Most risks manifest in privacy breaches and emerging threats, with a series of privacy concerns raising significant alarms. These risks include training data leakage, model reverse engineering, attribute inference attacks, model theft, and the paradox between memorization and generalization. In response to these risks, stringent data protection regulations have been enacted globally, imposing legal constraints on data processing activities. Examples include the General Data Protection Regulation (GDPR) in the EU, the Personal Information Protection Law (PIPL) in China, and the

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.LD28167

California Consumer Privacy Act (CCPA) in the U.S. These regulations indicate that mere anonymization of raw data may no longer suffice, as adversaries can exploit the aforementioned attack methods to infer personal information from the models themselves. Consequently, models have become a new critical point of privacy risk requiring stringent safeguards.

Failure to demonstrate compliance with privacy regulations exposes organizations to substantial fines and reputational damage. Driven by both technological and regulatory imperatives, it has become evident that protecting static, raw datasets alone is insufficient. The scope of protection must extend to dynamic AI models that encapsulate data information to mitigate emerging privacy breaches, ensure compliance, and rebuild user trust. As a result, the focus of privacy protection research has shifted from data security to computational process security, giving rise to a suite of privacy-preserving technologies such as differential privacy, federated learning, homomorphic encryption, and secure multi-party computation. This is not merely a technical challenge but a strategic imperative encompassing legal, ethical, and commercial dimensions.

Strengthening privacy protection safeguards fundamental human rights and personal dignity by preventing discrimination and bias while reducing security risks. Furthermore, it reinforces the core competitiveness and sustainable development of enterprises and institutions by preserving trade secrets, maintaining competitive advantages, and mitigating legal and public security risks while promoting the secure circulation and value realization of data assets.

Given the legal repercussions of privacy violations, robust privacy protection is imperative. From an ethical and social responsibility standpoint, ensuring the long-term development and societal acceptance of technology necessitates preventing the misuse of large language models and reinforcing privacy protection to fulfill corporate social responsibilities. Moreover, advancing privacy protection, sovereignty, and protecting critical infrastructure.

2. Privacy protection methodologies

2.1. Differential privacy

Privacy refers to information that individuals, organizations, or other entities are unwilling to disclose to external parties, such as personal salaries or medical records. While various protection frameworks based on anonymity and partitioning have emerged, differential privacy has been widely recognized as a more rigorous and robust protection model. The fundamental principle of this model involves introducing noise to raw data, its transformations, or statistical outputs to achieve privacy preservation. This method ensures that the insertion or deletion of any single record in a dataset will not affect the outcomes of any computational operations. Furthermore, this protection model is independent of an adversary's background knowledge; even if an attacker possesses information about all records except one, the privacy of that specific record remains uncompromised. The formal definition of differential privacy is as follows.

Definition 1. Given datasets D and D', which differ by at most one record (denoted as $|D \Delta D'| < 1$), and a privacy algorithm A with output range(A), if algorithm A satisfies the following inequality for any output result $o(o \in \text{Range}(A))$ on datasets D and D', then A satisfies $\varepsilon-$ differential privacy.

As can be inferred from Definition 1, differential privacy technology constrains the influence of any single record on the output results of algorithm A. This definition theoretically guarantees that algorithm A satisfies $\varepsilon-$ differential privacy, while the implementation of differential privacy protection necessitates the incorporation of a noise mechanism.

DOI: 10.54254/2755-2721/2025.LD28167

Differential privacy employs controlled noise perturbation to minimize the dependency of analytical results on any specific individual's data. By injecting carefully calibrated noise into data queries or model parameters, the mechanism prevents adversaries from inferring whether a particular individual participated in the original dataset by observing algorithmic outputs. Here, the privacy budget serves as the core parameter governing the strength of privacy protection [1].

The architectural framework of differential privacy for data protection comprises three components: the user terminal, the data manager, and the recipient. This framework supports flexible configurations of local and centralized modes based on the noise injection point, enabling differential privacy functionality in practical systems. The workflow is as follows:

(1) Local Differential Privacy Mode: The user terminal collects various types of data, performs necessary preprocessing, and injects noise while verifying compliance with local differential privacy requirements. The data is then aggregated via encrypted channels to the data manager for storage and transformation before being transmitted to the recipient for analysis.

(2) Centralized Differential Privacy Mode: After data collection and preprocessing at the user terminal, raw data is directly aggregated to the data manager, where noise is uniformly added in a centralized manner. The processed data is subsequently forwarded to the recipient for analysis. The distinction between these modes lies in the stage of noise injection and the specific algorithms employed.

Differential privacy has gained significant traction in the mobile internet industry. Applications based on the local differential privacy model—such as data collection, sharing, and analysis—are now widespread. Major technology firms, including Google, Apple, and Microsoft, have integrated localized differential privacy into their products, while companies like Uber and Volcano Engine leverage centralized differential privacy to safeguard user data in their platforms [2].

All differential privacy schemes adhere to the same principle: introducing noise to protect sensitive individual information. While higher noise levels ensure stronger privacy guarantees, lower noise increases the risk of sensitive data exposure. As discussed in Section II, evaluating privacy leakage against a given budget is essential before releasing datasets, machine learning models, or query responses [3].

2.2. Federated learning

Federated Learning (FL) is an advanced machine learning methodology that enables multiple devices or servers to collaboratively train a shared model without exchanging their locally stored data. Its core advantage lies in preserving data privacy and security by eliminating the need to transfer or replicate sensitive individual data to a central server. Based on data distribution among participants, FL is categorized into Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL).

HFL is particularly suitable for scenarios where data is distributed across numerous participants with aligned feature spaces. While different participants may possess distinct data samples, they share identical feature sets. For instance, in a multi-hospital medical research project analyzing electronic health records to improve pneumonia diagnosis accuracy, each hospital maintains structurally similar datasets (e.g., containing patient age, gender, symptoms, and test results) while serving different patient populations.

The FL-SZIDP framework adopts HFL architecture, leveraging distributed medical data and computational resources while ensuring patient data security and privacy. Suppose there are N healthcare institutions, each possessing data storage, processing, and transmission capabilities through multiple data collection devices, with local datasets Di. These institutions first train local

DOI: 10.54254/2755-2721/2025.LD28167

models using their patient data, then transmit model parameters (rather than raw data) to a central server for global model aggregation. This approach simultaneously mitigates the risk of original patient data exposure and reduces communication overhead.

To address security concerns in FL, differential privacy (DP) is incorporated as an additional protective layer. The proposed algorithm integrates DP with FL by injecting localized differential privacy noise into uploaded model parameters, thereby countering potential parameter leakage. While noise injection could theoretically occur at multiple stages, applying perturbations directly to participants' raw data might compromise data utility. Consequently, the framework strategically implements parameter-level noise perturbation after local training. This approach reduces individual data sensitivity and addresses privacy breaches potentially caused by honest-but-curious servers. Furthermore, the incorporation of identity authentication mechanisms effectively filters malicious participants to enhance overall data security [4].

2.3. Homomorphic encryption

Homomorphic encryption enables direct computation on ciphertexts, yielding results equivalent to processing plaintext before encryption. This allows data analysis without decryption, thereby ensuring data privacy. The technology is particularly suitable for cloud computing and big data scenarios, such as processing encrypted data directly in the cloud and returning valid results without exposing raw information. However, its high computational complexity creates significant efficiency bottlenecks when handling large-scale data, limiting widespread adoption.

In recent years, optimized approaches have improved practicality by reducing computational overhead, offering new perspectives for privacy-preserving computations in fields like healthcare and finance. These advancements gradually facilitate a balanced development between encryption technologies and efficient processing capabilities [5].

Different homomorphic encryption schemes vary in security requirements and computational capabilities. Therefore, a thorough comparison of their performance metrics and practical applicability is essential before selection. Currently, notable homomorphic encryption schemes include Paillier, BGV, and CKKS, each with distinct strengths and application scenarios.

The Paillier encryption scheme provides additive homomorphic functionality, enabling ciphertext addition operations. Its primary advantage lies in lower computational complexity, making it suitable for processing large volumes of additive ciphertext operations. For instance, with 32-bit integer operands, encryption takes approximately 0.3 ms, while decryption requires about 0.1 ms.

The BGV scheme is a fully homomorphic encryption (FHE) solution that supports unlimited addition and multiplication operations. A key feature is its use of bootstrapping to refresh ciphertexts, maintaining computational precision. For 32-bit integers, BGV encryption takes roughly 5 ms, decryption around 2 ms, while bootstrapping operations demand significantly longer durations, ranging from seconds to minutes.

The CKKS scheme is specifically designed for real-number arithmetic, supporting addition and multiplication on encrypted real numbers. Its primary advantage is floating-point compatibility, making it highly suitable for applications requiring complex mathematical functions and machine learning algorithms. Performance-wise, CKKS encryption averages 10 ms, with decryption taking approximately 5 ms.

Implementing homomorphic encryption in cloud environments requires ensuring both efficiency and security in the encryption process while meeting the scalability and manageability demands of cloud platforms. Critical considerations include key management, encryption algorithm configuration, and integration with cloud services. Therefore, deployment necessitates meticulous

DOI: 10.54254/2755-2721/2025.LD28167

planning, encompassing the selection of appropriate homomorphic encryption libraries, configuring key generation and distribution mechanisms, and optimizing encryption and decryption processes. Selecting a suitable homomorphic encryption library is fundamental to deployment [6].

Somewhat homomorphic encryption supports a predetermined number of homomorphic operations, limiting the allowed computational steps. Each operation increases underlying noise, meaning correct evaluation depends on performing only a bounded number of actions. Decryption fails when noise exceeds a certain threshold [7].

2.4. Secure multi-party computation

The history of secure multiparty computation (MPC) dates back several decades, tracing its origins to Yao's pioneering work, which was subsequently expanded upon by Goldreich, Micali, Wigderson, and numerous other scholars, leading to continuous advancements in this field. These studies have predominantly employed a similar methodological framework: first representing computational problems as combinatorial circuits, followed by executing concise protocols for each gate within the circuit by participating parties. While this approach is highly regarded for its universality and conceptual elegance, the resultant protocols inherently depend on circuit scale, which in turn is determined by the size of the input domain and the complexity of expressing such computations.

Historically, research in secure multiparty computation has primarily focused on theoretical exploration, with limited attention given to practical applications. The literature documents several MPC use cases, including private information retrieval (PIR), privacy-preserving statistical databases, and privacy-aware data mining [8].

Multiparty computation protocols enable a group of participants to collaboratively compute a function over their private inputs without disclosing any individual's data. Secure MPC specifically addresses scenarios where protocol execution may be compromised by adversarial attacks, wherein malicious actors might manipulate participants to extract private information or corrupt computational results. MPC protocols are designed to thwart such attacks, leveraging proven cryptographic techniques to ensure data privacy [9]. These secure protocols serve as the "brain" of proposed systems, allowing parties to jointly evaluate their data while preserving anonymity. They facilitate the computation of essential statistical measures—such as means, variances, or correlations—without revealing individual data inputs. Depending on specific survey requirements, additional MPC techniques may be employed, including secret sharing, homomorphic encryption, and Yao's garbled circuits [10].

3. Conclusion

This paper conducts a systematic and in-depth investigation into the privacy protection challenges encountered during the widespread application of large language models. Focusing on mainstream privacy-preserving computation techniques such as differential privacy, federated learning, homomorphic encryption, and secure multi-party computation, the study explores their applicability and implementation pathways throughout the data lifecycle of large language models, demonstrating substantial theoretical significance and practical value. The research not only provides a comprehensive analysis of potential privacy leakage risks across data collection, model training, and deployment phases, but also innovatively proposes a series of privacy protection strategies and evaluation frameworks designed to enhance data confidentiality and security while maintaining model performance. This work significantly contributes to the development of secure, trustworthy, and compliant large-scale AI systems by offering implementable technical solutions for enterprises

DOI: 10.54254/2755-2721/2025.LD28167

and developers, while simultaneously providing valuable references for legislative bodies and standard-setting organizations, thereby playing an active role in promoting the healthy and sustainable development of the artificial intelligence industry.

References

[1] Liu, Y., Jiang, C., Yang, S., et al. (2025). Differential privacy in federated learning: Current status, opportunities, and challenges. Journal of Frontiers of Computer Science and Technology, 1–29.

[2] Wang, Y., You, Y., Yuan, Q., et al. (2025). Applications of differential privacy in data protection. ZTE Communications, 1–8.

[3] El Ouadrhiri, A., & Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. IEEE Access, 10, 22359–22380.

[4] Cheng, Y., Jing, Y., Shi, Z., et al. (2025). A federated learning-based data privacy protection scheme. Computer and Modernization, (08), 63–69+88.

[5] Guan, Y., & Cao, Z. (2025). Research on data encryption technologies in information and communication under the context of big data. In Tianjin Electronics Industry Association (Ed.), Proceedings of the 2025 Annual Conference of Tianjin Electronics Industry Association (pp. 23–28). Tianjin 712 Communication & Broadcasting Co., Ltd.

[6] Liang, S. (2025). Application of homomorphic encryption algorithm in cloud storage data security. Cybersecurity and Information Technology, (04), 136–138.

[7] Pulido-Gaytan, B., et al. (2021). Privacy-preserving neural networks with homomorphic encryption: Challenges and opportunities. Peer-to-Peer Networking and Applications, 14(3), 1666–1691.

[8] Du, W., & Atallah, M. J. (2001). Secure multi-party computation problems and their applications: A review and open problems. In Proceedings of the 2001 Workshop on New Security Paradigms (pp. 13–22).

[9] Li, X., Dowsley, R., & De Cock, M. (2021). Privacy-preserving feature selection with secure multiparty computation. In International Conference on Machine Learning (pp. 6326–6336). PMLR.

[10] Alghamdi, W., Salama, R., Sirija, M., et al. (2023). Secure multi-party computation for collaborative data analysis. In E3S Web of Conferences (Vol. 399, p. 04034). EDP Sciences.