Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.076358

REVIEW

A Review on Penetration Testing for Privacy of Deep Learning Models

Salma Akther[1], **Wencheng Yang**[1,*], **Song Wang**[2], **Shicheng Wei**[1], **Ji Zhang**[1], **Xu Yang**[3], **Yanrong Lu**[4] **and Yan Li**[1]

[1]School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, QLD, Australia
[2]Department of Engineering, La Trobe University, Melbourne, VIC, Australia
[3]School of Computer and Data Science, Minjiang University, Fuzhou, China
[4]School of Security Science and Engineering, Civil Aviation University of China, Tianjin, China

*Corresponding Author: Wencheng Yang. Email: wencheng.yang@unisq.edu.au

Received: 19 November 2025; Accepted: 22 January 2026; Published: 12 March 2026

ABSTRACT: As deep learning (DL) models are increasingly deployed in sensitive domains (e.g., healthcare), concerns over privacy and security have intensified. Conventional penetration testing frameworks, such as OWASP and NIST, are effective for traditional networks and applications but lack the capabilities to address DL-specific threats, such as model inversion, membership inference, and adversarial attacks. This review provides a comprehensive analysis of penetration testing for the privacy of DL models, examining the shortfalls of existing frameworks, tools, and testing methodologies. Through systematic evaluation of existing literature and empirical analysis, we identify three major contributions: (i) a critical assessment of traditional penetration testing frameworks' inadequacies when applied to DL-specific privacy vulnerabilities, (ii) a comprehensive evaluation of state-of-the-art privacy-preserving methods and their integration with penetration testing workflows, and (iii) the development of a structured framework that combines reconnaissance, threat modeling, exploitation, and post-exploitation phases specifically tailored for DL privacy assessment. Moreover, this review evaluates popular solutions such as IBM Adversarial Robustness Toolbox and TensorFlow Privacy, alongside privacy-preserving techniques (e.g., Differential Privacy, Homomorphic Encryption, and Federated Learning), which we systematically analyze through comparative studies of their effectiveness, computational overhead, and practical deployment constraints. While these techniques offer promising safeguards, their adoption is hindered by accuracy loss, performance overheads, and the rapid evolution of attack strategies. Our findings reveal that no single existing solution provides comprehensive protection, which leads us to propose a hybrid approach that strategically combines multiple privacy-preserving mechanisms. The findings of this survey underscore an urgent need for automated, regulation-compliant penetration testing frameworks specifically tailored to DL systems. We argue for hybrid privacy solutions that combine multiple protective mechanisms to ensure both model accuracy and privacy. Building on our analysis, we present actionable recommendations for developing adaptive penetration testing strategies that incorporate automated vulnerability assessment, continuous monitoring, and regulatory compliance verification.

KEYWORDS: Penetration testing; deep learning; homomorphic encryption; differential privacy; federated learning

1 Introduction

The systematic process of identifying and analyzing security vulnerabilities within a system, network, or application to uncover weaknesses before malicious actors can exploit them is known as Penetration Testing, also known as Pen Testing or Ethical Hacking [1,2]. Penetration testing has become essential for cybersecurity because it enables industries and companies to proactively identify security gaps and strengthen their defensive measures against cyber threats [3–5]. The primary goal of penetration testing is to discover and evaluate security flaws in a system before malicious actors can leverage them for unauthorized access or data exfiltration [6]. This process typically involves simulating various attack scenarios, evaluating the system's response, and providing actionable recommendations to mitigate identified vulnerabilities [7–9]. Through systematic penetration testing, businesses can effectively prioritize and remediate vulnerabilities, which significantly reduces the risk of data breaches, information theft, financial damage, and reputational harm.

Deep Learning (DL), a specialized branch of machine learning, has fundamentally transformed numerous industries through its ability to process and analyze vast quantities of data. DL excels in domains requiring complex pattern recognition, such as image identification, predictive analytics, and natural language processing, which leverage multi-layered neural networks [10]. These sophisticated neural architectures enable advanced decision-making capabilities that have revolutionized critical sectors [11], including healthcare (e.g., disease diagnosis [12–14], medical image analysis [15–17], e-health records analysis [18–20]), banking (e.g., algorithmic trading, fraud detection, risk management [21,22]), autonomous systems (e.g., robotics, self-driving automobiles [23,24]), and identity management(e.g., biometric recognition [25–28], identity privacy protection [29–31]). Despite these advances, such platforms inherently process and store sensitive data, which renders them attractive targets for malicious attacks. The ubiquitous deployment of DL systems has consequently generated substantial privacy concerns.

While prior research has examined individual DL privacy threats including model inversion, model extraction, membership inference attacks, data poisoning, and adversarial exploitation, such studies typically analyze each attack separately and therefore do not provide a holistic evaluation of a deployed model's privacy posture [6]. Traditional privacy-attack analyses often demonstrate the feasibility of specific attacks under controlled laboratory settings, without considering how these threats manifest across the full lifecycle of real-world systems [32]. In practice, DL models can leak private information through outputs, gradients, confidence scores, and latent representations, making them vulnerable in ways fundamentally different from classical software applications. These diverse leakage pathways underscore the need for more rigorous and comprehensive privacy-evaluation methodologies.

Penetration testing (PT) is therefore essential because it offers a structured, multi-phase, adversarial methodology capable of evaluating privacy leakage under realistic operational conditions. Unlike standalone attack demonstrations, PT incorporates reconnaissance, threat modeling, exploitation, and post-exploitation analysis, enabling a full-spectrum understanding of how privacy vulnerabilities emerge, propagate, and interact across interconnected DL components. Furthermore, PT evaluates practical deployment specific factors such as API exposure, query limitations, gradient access, and system integration that traditional privacy-attack research often overlooks. This distinction highlights why PT is critical for assessing DL privacy and why existing privacy analyses alone are insufficient for securing real-world deployments.

This review addresses a critical gap in the literature by being the comprehensive analysis to specifically examine penetration testing methodologies for DL privacy, which distinguishes it from previous surveys that have either focused solely on general DL security [10] or traditional penetration testing without DL considerations [4]. The motivation for this review arises from the recognition that existing privacy-attack research does not systematically incorporate attacker behaviour, lifecycle-based evaluation, or operational constraints, all of which are central components of penetration testing. While prior research on DL-related security risks and privacy protection has established foundational understanding, it has not systematically evaluated how traditional penetration testing can be adapted for DL-specific privacy threats. Through comprehensive integration of disparate resources, this review aims to establish a unified framework for developing systematic, reliable penetration testing strategies that specifically target the privacy vulnerabilities inherent in DL systems. The comprehensive analytical framework of this review is illustrated in Fig. 1. Fig. 1 represents a conceptual and methodological framework that integrates traditional penetration testing principles with deep learning–specific privacy threats and defenses. The components shown in the figure (e.g., privacy threats, evaluation objectives, and lifecycle-based assessment stages) are iterative and interdependent, and can be revisited or adapted depending on the model architecture, threat assumptions, and evaluation objectives. This framework serves as a systematic guideline for designing, conducting, and interpreting privacy penetration testing in deep learning systems.

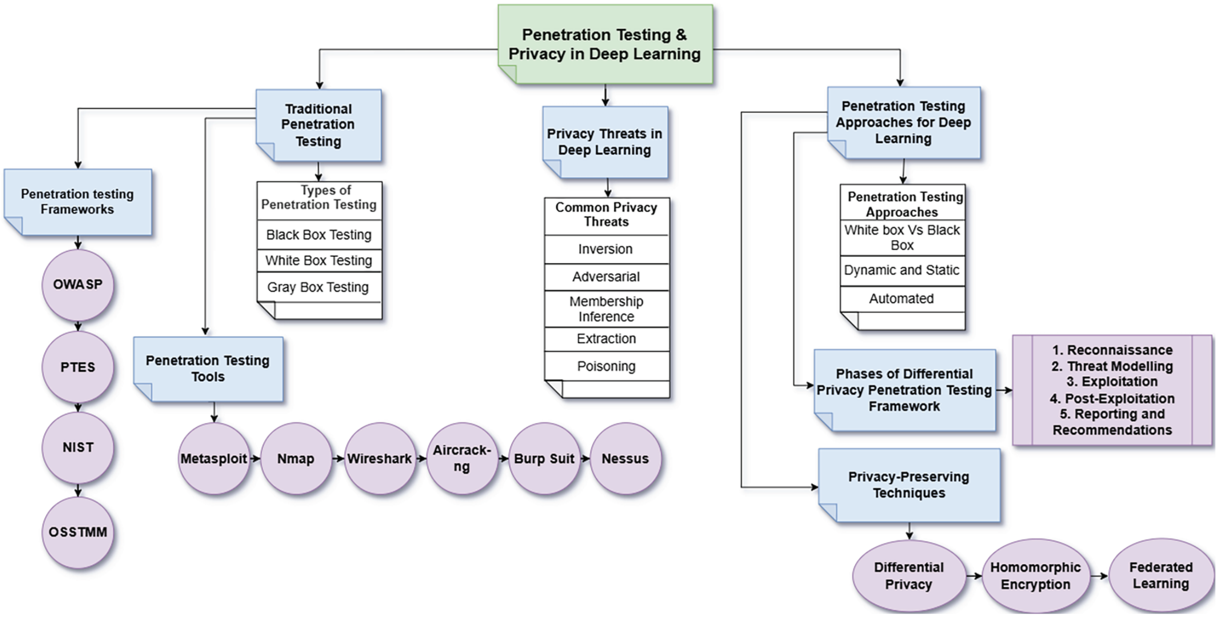
Penetration Testing & Privacy in Deep Learning
Traditional Penetration Testing
Penetration testing Frameworks
OWASP
PTES
NIST
OSSTMM
Types of Penetration Testing
Black Box Testing
White Box Testing
Gray Box Testing
Penetration Testing Tools
Metasploit
Nmap
Wireshark
Aircrack-ng
Burp Suit
Nessus
Privacy Threats in Deep Learning
Common Privacy Threats
Inversion
Adversarial
Membership Inference
Extraction
Poisoning
Penetration Testing Approaches for Deep Learning
Penetration Testing Approaches
White box Vs Black Box
Dynamic and Static
Automated
Phases of Differential Privacy Penetration Testing Framework
1. Reconnaissance
2. Threat Modelling
3. Exploitation
4. Post-Exploitation
5. Reporting and Recommendations
Privacy-Preserving Techniques
Differential Privacy
Homomorphic Encryption
Federated Learning

Figure 1: Overall workflow diagram illustrating the penetration testing framework for deep learning privacy.

This survey aims to fill this gap by providing a *comprehensive review of penetration testing for DL privacy*. Specifically, the contributions of this work are fourfold:

1. A critical analysis of the limitations of traditional penetration testing frameworks when applied to DL privacy.
2. A taxonomy of DL privacy threats and their mapping to penetration testing phases.
3. A comparative evaluation of state-of-the-art privacy-preserving methods and associated tools (e.g., IBM ART, TensorFlow Privacy, PySyft, FATE).
4. Actionable recommendations and a future research agenda for hybrid, automated, and regulation-compliant penetration testing tailored to DL systems.

Unlike existing DL privacy and federated learning surveys, this review integrates privacy attacks, defenses, and evaluation within a penetration testing life-cycle tailored to deep learning systems, which is not jointly addressed in prior surveys. The comparison of our survey with some related existing surveys is listed in Table 1. The comparison highlights differences in scope and methodological focus rather than overall completeness or depth.

Table 1: Comparison of prior related surveys and this review.

Survey	DL Privacy Attacks	Privacy-Preserving Techniques	PT Perspective	PT Lifecycle Mapping	DL-Specific PT Workflow	Risk Evaluation Framework
Alwabisi [33]	Partial	No	Yes	No	No	No
Zhang et al. [34]	Yes	Yes	No	No	No	No
Thapa et al. [35]	No	Yes	No	No	No	No
Chen et al. [36]	Partial	Yes	No	No	No	No
Nasr et al. [37]	Yes	No	No	No	No	No
Kairouz et al. [38]	Partial	Yes	No	No	No	No
This Survey	Yes	Yes	Yes	Yes	Yes	Yes

In contrast to AI red-teaming studies, which typically evaluate adversarial behaviours in isolated or robustness-focused scenarios, this review adopts a formal penetration testing life-cycle with explicit consideration of privacy leakage, system-level impact, and regulatory compliance for deep learning models.

On the above table, “Yes” indicates explicit coverage; “Partial” denotes limited discussion; “No” indicates that the aspect is not addressed.

Literature Review Methodology

This review adopts a structured literature analysis to identify relevant studies on penetration testing and deep learning privacy systems. Primary databases including IEEE Xplore, ACM Digital Library, ScienceDirect, Scopus, and Google Scholar. The search process utilised a combination of targeted keywords and logical operators to retrieve relevant studies. Core search terms included deep learning privacy, penetration testing, Penetration testing in deep learning, model inversion, membership inference, adversarial attacks, differential privacy, homomorphic encryption, federated learning and privacy-preserving machine learning. These terms were iteratively refined to ensure broad coverage while maintaining relevance to the research scope.

The inclusion criteria comprised peer-reviewed journal articles, conference papers, and authoritative technical reports, as well as preprint repositories were consulted to capture recent and emerging research developments focusing on DL privacy threats, penetration testing methodologies, or privacy-preserving techniques. Studies unrelated to machine learning, lacking security relevance, or focused solely on traditional network security were excluded. Reference management and duplicate removal were conducted using Zotero to ensure consistency. Following the screening titles, abstracts, and full texts, more than 80 peer-reviewed studies were selected for in-depth analysis. The selected literature was then organized into thematic categories covering deep learning privacy threats, penetration testing strategies, privacy-preserving mechanisms, evaluation tools, and emerging research challenges. This systematic selection process supports the comprehensive and balanced nature of the review.

The remainder of this paper follows a structured progression from foundational concepts to practical applications and future directions. Section 2 establishes the theoretical foundation by examining traditional penetration testing, with particular emphasis on key methodologies, including black-box testing, white-box testing, and gray-box testing, alongside critical evaluation of established penetration testing frameworks including OWASP, PTES, NIST, and OSSTMM. Section 3 provides a comprehensive taxonomy of privacy risks specific to deep learning models, encompassing model inversion, adversarial attacks, membership inference, model extraction, and data poisoning. Drawing from these identified privacy vulnerabilities, Section 4 investigates the adaptation and enhancement of penetration testing methodologies to effectively evaluate and strengthen DL privacy, which includes detailed analysis of targeted testing strategies, attack simulation tools, and the integration of privacy-preserving techniques such as differential privacy, homomorphic encryption, and federated learning. Section 5 articulates the research questions that frame this investigation, while Section 6 critically analyzes existing challenges and presents future research directions. Section 7 concludes this work by synthesizing the key findings and contributions of this comprehensive review.

2 Traditional Penetration Testing

The evolving complexity of modern technologies and the increasing sophistication of attack vectors drive continuous changes in cybersecurity risks. Penetration testing stands as one of the most widely adopted methodologies for analyzing security vulnerabilities, which replicates real-world attacks to evaluate a system's security posture through assessment of configuration errors, weak passwords, insecure coding practices, and unpatched software. This testing methodology constitutes an essential component of proactive security frameworks (ISO 27001, NIST 800-53) [39–42] to ensure organizational compliance with security standards and regulatory requirements. The process involves deploying targeted attacks against the system to evaluate its security controls and determine the system's resilience against potential threats [43]. Even though traditional penetration testing methods are extensively employed in information technology security, their effectiveness remains constrained when addressing privacy-specific vulnerabilities in DL models.

The following sections provide an examination and analysis of penetration testing types, tools, and frameworks.

2.1 Types of Penetration Testing

Penetration testing techniques are generally classified into three distinct categories (shown in Fig. 2, which is adapted from [44]) based on the tester's level of system knowledge [45,46]. These classification frameworks are well established in traditional cybersecurity assessments, their direct applicability to deep learning (DL) privacy evaluation is limited, as they do not inherently capture model-specific privacy leakage or training-data exposure.

In **Black-Box Testing**, testers possess no prior knowledge regarding the target system's internal architecture simulating real-world external attackers who rely on reconnaissance and observable system behaviour [39,47,48]. The methodology proves particularly effective for evaluating the security posture of organization-facing systems, such as firewalls, web applications, and public APIs (Application Programming Interface). Despite its effectiveness in simulating genuine attack vectors, provides limited visibility into DL model internals and cannot directly assess privacy leakage arising from learned representations.

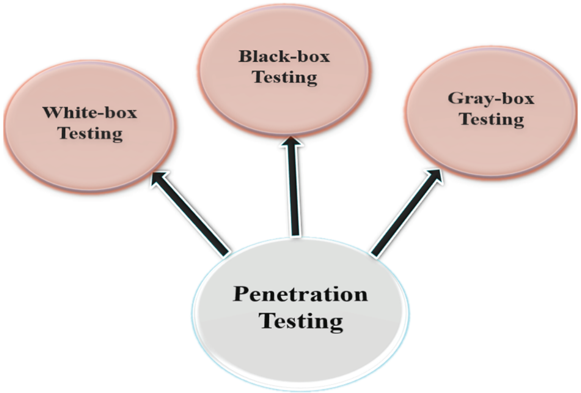

Figure 2: Three types of penetration testing.

In **White-Box Testing**, alternatively referred to as clear-box testing, glass-box testing, structural-box testing, and transparent-box testing [47–49], grants testers full access to system architecture, source code, and configurations, enabling detailed identification of design flaws and implementation weaknesses. This methodology resembles internal security audits, wherein testers examine all system components, such as access control configurations, encryption implementations, and programmatic logic. However, this method incurs higher operational costs and demands extensive data requirements.

Gray-Box Testing, also designated as translucent-box testing, provides testers with partial information about the target system. This scenario simulates conditions where attackers possess limited insider knowledge, such as compromised credentials or restricted Application Programming Interface (API) access [47,49]. Gray-box testing enables testers to focus on high-risk areas without the extensive resource requirements of white-box testing, thus representing a strategic balance between effectiveness and operational authenticity. Nevertheless, without ML-specific evaluation techniques, gray-box testing remains insufficient for systematically identifying deep learning privacy threats such as membership inference or model inversion.

From an overall perspective, although these testing models remain foundational to penetration testing, their applicability to DL privacy assessment is limited unless extended with DL-specific attack methodologies and evaluation metrics.

2.2 Penetration Testing Tools

A range of established penetration testing tools is routinely used to assess system security across diverse technological environments. The following examination presents some of the most widely adopted and effective tools in contemporary penetration testing practices. However, these tools are primarily designed for network, application, and protocol-level vulnerabilities and offer limited support for evaluating DL-specific privacy risks. Below a comparison of these penetration testing tools is listed in Table 2.

Table 2: Comparison of traditional penetration testing tools mentioned in the survey.

Tool	Primary Function	Strengths	Limitations for DL Privacy
Metasploit	Exploitation framework	Extensive exploit database; automated payload generation; post-exploitation modules for simulating advanced persistent threats.	Highly effective in traditional network and system penetration testing, but lacks modules for adversarial attacks or DL-specific privacy risks.
Nmap	Network scanning and reconnaissance	Identifies active hosts, services, open ports; supports custom scripts via Nmap Scripting Engine (NSE).	Provides valuable network-level intelligence, but cannot detect DL vulnerabilities such as model inversion or membership inference.
Wireshark	Network traffic analyser	Captures and inspects packets in real time; supports hundreds of protocols; enables forensic traffic analysis.	Limited to communication-level threats; does not address adversarial perturbations, data poisoning, or DL-specific leakages.
Aircrack-ng	Wireless security testing	Supports WEP/WPA key recovery; includes packet capture and cryptographic attack utilities.	Specialised for wireless networks; offers no testing capabilities for DL privacy vulnerabilities.
Burp Suite	Web application penetration testing	Detects SQL injection, cross-site scripting (XSS), and authentication bypasses; provides interception proxy and automated scanning.	Focuses on application-layer flaws; does not evaluate DL models' resilience to inference attacks or adversarial examples.
Nessus	Vulnerability scanner	Automated detection of misconfigurations, default credentials, missing patches; supports compliance reporting.	Enterprise-grade scanner, but lacks specialised functions to test DL models against membership inference, model extraction, or poisoning attacks.

Metasploit represents a robust and comprehensive penetration testing framework that supports automated exploitation, payload delivery, and post-exploitation analysis through an extensive library of attack modules. This framework serves as a flexible platform for replicating diverse attack scenarios and provides automated exploitation capabilities [11,50]. The tool incorporates trailblazing features such as post-exploitation modules, payload encoding mechanisms, and evasion techniques that allow testers to simulate advanced persistent threats. Metasploit proves particularly beneficial for discovering security vulnerabilities in network services and evaluating the effectiveness of existing security policies through controlled exploitation attempts. However, despite its strengths in traditional network and system security assessment, Metasploit does not provide native support for evaluating deep learning–specific threats, such as adversarial attacks, membership inference, or privacy leakage arising from model inference behavior.

Nmap (Network Mapper) functions as a sophisticated network scanning tool that facilitates comprehensive network reconnaissance through identification of active hosts, running services, open ports, and operating system fingerprinting, which collectively provide valuable intelligence for identifying potential attack vectors [11,51]. Its scripting engine further supports service enumeration and vulnerability detection across large-scale infrastructures. However, while Nmap is effective for mapping network-level attack surfaces, it does not support evaluation of deep learning–specific privacy risks, such as inference-based data leakage or adversarial manipulation of model behaviour.

Wireshark operates as an advanced network protocol analyzer that assists security experts in identifying suspicious network traffic patterns through real-time packet capture and detailed protocol analysis [11,52]. The application provides deep packet inspection capabilities that enable forensic analysis of network communications, protocol violations, and anomalous traffic behaviors [53]. It is commonly employed for traffic monitoring and post-incident forensic analysis in traditional network security assessments. However, Wireshark operates at the communication layer and does not provide visibility into deep learning model behavior or support assessment of privacy risks arising from inference outputs, gradient leakage, or adversarial manipulation.

Aircrack-ng comprises a comprehensive suite of tools specifically designed for wireless network security assessment, network monitoring, and WEP/WPA key recovery operations, which includes specialized packet capture utilities and cryptographic attack implementations [11,51]. It is commonly used to assess vulnerabilities in wireless configurations through passive monitoring and active attack techniques. However, Aircrack-ng focuses exclusively on wireless communication security and does not support evaluation of deep learning–specific privacy risks or attacks targeting model inference or training processes.

Burp Suite serves as a specialized web application security testing platform that facilitates detection of application-layer vulnerabilities through integrated features supporting session management analysis, automated scanning, and manual testing capabilities [54]. The platform excels at identifying common web application vulnerabilities such as cross-site scripting (XSS), SQL injection, and authentication bypass vulnerabilities. Burp Suite's proxy functionality enables testers to intercept, modify, and analyze HTTP/HTTPS traffic to identify security weaknesses in web application logic and input validation mechanisms. However, Burp Suite is limited to web application logic and does not provide mechanisms for evaluating deep learning–specific privacy risks, including inference-based data leakage or adversarial manipulation of model outputs.

Nessus functions as an enterprise-grade vulnerability scanner that systematically identifies potential attack vectors, security misconfigurations, and system vulnerabilities through automated assessment protocols [11,55]. The scanner employs comprehensive vulnerability databases and performs automated assessments to identify weaknesses such as default credentials, misconfigured systems, and missing security patches. It is effective for prioritizing remediation based on exploitability and compliance requirements. However, this established vulnerability scanner lacks the specialized capabilities required for assessing DL-specific security risks, such as adversarial perturbations or data poisoning attacks that target machine learning model integrity.

Despite their proven effectiveness and widespread adoption in traditional cybersecurity contexts, these conventional penetration testing tools are not optimally suited for addressing the unique security challenges and privacy concerns associated with DL-related system assessments. The emergence of AI-specific vulnerabilities necessitates the development of specialized testing frameworks and tools designed specifically for deep learning model security evaluation.

2.3 Penetration Testing Frameworks

The scoring in Table 3 is derived from documented capabilities in each framework's official guidelines, including the OWASP Testing Guide and ASVS, PTES Technical Guidelines, NIST SP 800-53, the NIST AI Risk Management Framework, and OSSTMM 3. Each framework was evaluated against five DL-relevant criteria: (1) coverage of DL-specific privacy threats (e.g., inversion, extraction, membership inference), (2) support for ML pipeline assessment, (3) alignment with structured attack life-cycle methodologies, (4) mechanisms for evaluating privacy-leakage behaviors, and (5) regulatory alignment. Higher scores were assigned when a framework demonstrated explicit support for AI/ML security considerations or model-centric testing procedures; lower scores reflect limited or no applicability to DL privacy risks. This structured scoring method ensures transparency, reproducibility, and direct relevance to DL-privacy assessment.

Table 3: Evaluation of traditional penetration testing frameworks for deep learning privacy.

Framework	DL Privacy Threat Coverage	ML Pipeline Testing	Attack Life-Cycle Integration	Privacy Leakage Analysis	Regulatory Compliance
OWASP	•	○	••	○	•••
PTES	••	•	••••	○	••
NIST	•	○	•••	○	••••
OSSTMM	••	○	•••	○	••

Note: •••• = Strong capability; ••• = Good capability; •• = Moderate capability; • = Limited capability; ○ = No capability.

OWASP (Open Web Application Security Project) provides extensive methodologies for web and application penetration testing, including vulnerability identification, authentication failures, and configuration weaknesses [44,56]. However, OWASP offers minimal support for evaluating model-centric vulnerabilities and lacks mechanisms for analyzing privacy leakage in DL models, such as training data exposure, gradient-based attacks, or inference-time exploitation. Its partial alignment with the PT attack life-cycle through structured testing phases such as threat enumeration, exploitation, and reporting explains its moderate score for life-cycle integration, while its strong emphasis on compliance-oriented security practices supports its high regulatory rating [57,58]. Despite OWASP's proven effectiveness for traditional application security evaluations, it does not address specialized DL-specific vulnerabilities, including model inversion, membership inference, and adversarial perturbation attacks that specifically target neural-network integrity. Consequently, OWASP remains a strong framework for conventional application security but provides only limited applicability to DL privacy risks, reflected in its low scores for DL privacy-threat coverage, ML pipeline testing, and privacy leakage analysis.

PTES (Penetration Testing Execution Standard) defines a highly structured seven-phase methodology, ranging from intelligence gathering to exploitation and reporting. This structured workflow aligns well with adversarial evaluation processes [59]. Although PTES is valuable for modelling attacker behaviour, it does not address DL-specific vulnerabilities such as gradient leakage, shadow modelling, or model-extraction techniques. Consequently, it receives a high score for lifecycle support but only moderate relevance for DL privacy assessments [4].

The **NIST (National Institute of Standards and Technology)** provides comprehensive compliance-oriented frameworks, including SP 800-53 and the NIST AI Risk Management Framework. These guidelines incorporate data-governance principles and acknowledge adversarial machine-learning risks [4,54]. However, NIST does not define penetration testing procedures for identifying privacy leakage in DL models or assessing model-specific attack vectors. Therefore it scores strongly on regulatory alignment but only moderately on DL relevance.

The **OSSTMM (Open-Source Security Testing Methodology Manual)** provides a quantitative, metrics-driven approach to security testing and is widely recognized for its structured and scientifically grounded assessment methodology. While OSSTMM is highly effective for evaluating organizational security across traditional domains, such as operational processes, physical controls, and data handling, it does not include specialized assessment criteria for analyzing machine-learning systems [44,54]. However, OSSTMM lacks mechanisms for analysing ML pipelines or identifying privacy leakage pathways such as model inversion or gradient exposure. As a result, although it performs reasonably well in structured security measurement, its applicability to DL privacy testing remains limited [60].

2.4 DL/ML-Specific Tools

This subsection highlights DL/ML-specific security tools that complement traditional penetration testing by enabling direct evaluation of deep learning privacy threats that conventional tools cannot assess. To address the shortcomings of traditional penetration testing frameworks, it is necessary to incorporate DL/ML-specific security toolkits that directly target vulnerabilities unique to deep learning systems [61]. In particular, libraries such as IBM ART [62,63] and CleverHans provide comprehensive capabilities for generating adversarial examples, evaluating model robustness, and conducting membership inference and model inversion attacks—capabilities that are specifically designed for neural network architectures. Similarly, TensorFlow Privacy and Opacus [64] enable rigorous auditing of differential-privacy guaranties during training, offering mechanisms to quantify and monitor privacy leakage. Together, these tools complement conventional PT methodologies by allowing a systematic evaluation of DL-specific attack surfaces that traditional cybersecurity frameworks are not designed to assess.

Although the tools discussed in this subsection primarily support adversarial testing and attack simulation, complementary DL/ML-specific security tools that enable defensive evaluation of privacy-preserving mechanisms, such as differential privacy, homomorphic encryption, and federated learning, are examined in Section 4.3 within a penetration testing life-cycle context.

Unlike Table 2, which highlights the limitations of conventional penetration testing tools for deep learning privacy assessment, Table 4 emphasizes the penetration testing relevance of DL/ML-specific security tools that directly support adversarial testing, privacy leakage evaluation, and defensive validation.

Table 4: Comparison of DL/ML-specific security tools for privacy-oriented penetration testing.

Tool	Primary Purpose	Strengths	Penetration Testing Relevance
IBM Adversarial Robustness Toolbox (ART)	Adversarial ML security testing	Generates adversarial examples; supports membership inference, model inversion, model extraction, and poisoning attacks; evaluates model robustness.	Enables systematic exploitation-phase testing of DL privacy and robustness vulnerabilities aligned with penetration testing objectives.

(Continued)

Table 4 (continued)

Tool	Primary Purpose	Strengths	Penetration Testing Relevance
CleverHans	Adversarial attack simulation	Implements state-of-the-art adversarial attacks (e.g., FGSM, PGD); supports evaluation of model sensitivity to crafted inputs.	Supports exploitation-phase testing by simulating realistic adversarial behaviors against DL models.
TensorFlow Privacy	Differential privacy auditing	Implements DP-SGD; provides privacy accounting; enables monitoring of privacy loss during training.	Supports post-exploitation and evaluation phases by quantifying residual privacy leakage under attack.
Opacus	Differential privacy for PyTorch	Privacy accounting; per-sample gradient clipping; DP training for deep models.	Enables defensive evaluation of privacy-preserving mechanisms within a penetration testing lifecycle.

3 Privacy Issues of DL Models

DL model depends on massive datasets, many of which include private or sensitive data [9]. Despite their success, DL models have the potential to unintentionally remember and disclose details about training data, which raises serious privacy issues. This is particularly concerning when there is personally identifiable information or other sensitive data gathered in training. As DL models grow exponentially and are used in more delicate fields, it is crucial to address these privacy issues. To ensure relevance to penetration testing methodologies, the privacy attacks discussed in this section are later mapped to standard penetration testing phases in Section 3.2.

3.1 Common Privacy Threats

This section presents some common privacy threats to DL models. The severity levels, likelihood of occurrence, and potential impact of these attacks are systematically compared in Fig. 3, which provides a comprehensive risk assessment framework for understanding the relative threat landscape facing modern deep learning deployments. In Fig. 3, risk severity is assessed based on three core dimensions: (i) attack feasibility, reflecting the technical effort required to execute the attack; (ii) privacy impact, capturing the extent of sensitive information exposure; and (iii) regulatory and operational consequences, particularly in compliance-sensitive domains. The scoring criteria are informed by established security risk assessment methodologies, including OWASP risk rating principles, NIST SP 800-53 controls, and prior literature on deep learning privacy attacks. Each dimension is evaluated using an ordinal scale, facilitating consistent and reproducible comparison of privacy risks across different attack categories [65,66].

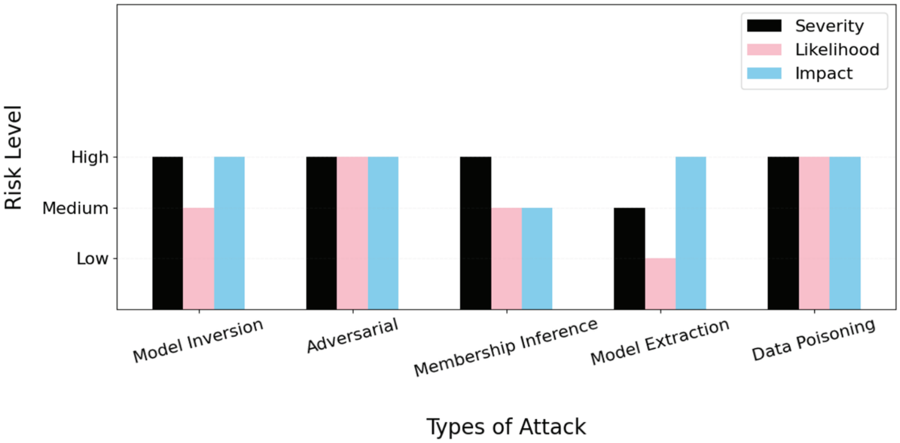

Figure 3: Privacy attack risk comparison regarding DL models.

Model Inversion Attacks. In this sophisticated category of attack, adversaries leverage the outputs and behavioral patterns of established DL models to systematically retrieve sensitive input data that was used during the training process [67]. Through strategic use of publicly accessible data and carefully crafted queries to DL model prediction interfaces, attackers can successfully reconstruct significant portions of the training database and recover highly private information, such as patient health records in medical diagnostic applications or facial images in biometric authentication systems. Furthermore, DL models that process biometric data or personally identifiable information demonstrate particular vulnerability to this attack vector due to the inherent memorization properties of deep neural networks [39]. Attackers systematically exploit the built-in capacity of DL models to retain and encode data-specific characteristics within their learned representations, making robust privacy protection measures absolutely critical for mitigating these substantial associated risks. The success of model inversion attacks often depends on the model's complexity, the amount of training data, and the sensitivity of the information encoded within the model parameters.

Adversarial Attacks. In this category of sophisticated manipulation, attackers deliberately alter input data such as text, audio, or image content to deceive DL models while maintaining the appearance of legitimate data to human observers [39]. During the process of strategically modifying input data, attackers cause DL models to generate incorrect predictions or classifications that appear normal to human reviewers, thereby compromising system reliability and decision-making processes. Beyond causing functional failures, these attacks can create significant security vulnerabilities that enable confidentiality breaches through disclosure of trained data patterns and internal model information. Adversarial attacks are systematically categorized into two primary methodologies: (1) black-box attacks, where attackers interact with the system without prior knowledge of its internal architecture or parameters; and (2) white-box attacks, where attackers possess comprehensive access to the model's structural properties, weights, and training procedures. Consequently, mission-critical applications that depend on reliable security decisions, such as unmanned vehicle navigation systems and electronic health platforms, face particularly severe vulnerabilities to adversarial attack vectors that could result in catastrophic failures.

Membership Inference Attacks. In this privacy-focused attack methodology, adversaries can systematically determine whether specific information of interest was included in a model's training dataset through careful analysis of model responses and behavioral patterns [37,39,68]. Attackers employ carefully selected data samples to query DL models systematically, then analyze the model's responses, confidence levels, and prediction patterns to determine whether particular datasets were utilized during the training process. For example, malicious actors could identify whether a specific patient's health records were utilized to train a medical diagnostic system, thereby revealing sensitive information about individual participation in research studies or clinical trials. These privacy violations create substantial concerns in highly regulated sectors such as healthcare and financial services, where unauthorized disclosure of sensitive or confidential information carries significant legal consequences and financial penalties for organizations.

Model Extraction Attacks. In this intellectual property-focused attack vector, adversaries systematically and repeatedly interrogate a deployed DL model through its public interface to reverse-engineer, recreate, or approximate its functionality without requiring direct access to the original training data or model architecture [39]. Through this systematic extraction process, attackers can retrieve confidential algorithmic information, steal valuable intellectual property, or create unauthorized replications that enable further malicious activities. The extracted model copies can then be used for competitive advantage, sold to unauthorized parties, or serve as a foundation for developing more sophisticated attacks against the original system or similar deployments.

Data Poisoning Attacks. In this supply chain-focused attack methodology, malicious actors deliberately inject corrupted, biased, or strategically modified data into the original training dataset to fundamentally alter the system's behavior, thereby degrading performance or creating hidden backdoors that enable future exploitation [39]. This attack vector poses particularly severe threats to DL models whose operational effectiveness depends heavily on user-generated content or crowdsourced data, such as fraud detection systems, content moderation platforms, or recommendation engines. The injected poisoned data can remain dormant until specific trigger conditions are met, making detection extremely challenging and enabling long-term compromise of system integrity and reliability.

3.2 Mapping Deep Learning Privacy Attacks to Penetration Testing Phases

While the privacy attacks discussed in Section 3 are well-established threats to deep learning (DL) models, their relevance to penetration testing lies in their evaluation within a structured adversarial lifecycle. Accordingly, these attacks are mapped to traditional penetration testing phases— reconnaissance, exploitation, and post-exploitation, to enable systematic and reproducible DL privacy assessments [33].

Model inversion attacks primarily occur during reconnaissance and exploitation, where exposed prediction APIs and confidence scores are identified and leveraged to reconstruct sensitive training data. Post-exploitation analysis quantifies privacy leakage and assesses confidentiality violations.

Adversarial attacks are typically executed during the exploitation phase by injecting crafted perturbations that induce incorrect model behavior. Reconnaissance focuses on understanding input constraints and decision boundaries, while post-exploitation evaluates system reliability and safety impacts.

Membership inference attacks span all penetration testing phases, beginning with reconnaissance of output patterns, followed by exploitation through systematic querying, and post-exploitation assessment of regulatory and privacy implications.

Model extraction attacks align with iterative reconnaissance and exploitation, where repeated queries are used to approximate or replicate deployed models. Post-exploitation evaluates intellectual property exposure and risks of secondary attacks.

Data poisoning attacks are primarily assessed during exploitation, targeting training or update pipelines. Reconnaissance identifies data ingestion and trust assumptions, while post-exploitation examines backdoor persistence and long-term integrity degradation.

By explicitly mapping DL privacy attacks to penetration testing phases, this section demonstrates how penetration testing enables structured evaluation and mitigation of privacy risks throughout the DL system life-cycle.

3.3 PT-Oriented Privacy Attacks and Defences in Large Models

Recent advances in large-scale deep learning models, particularly large language models (LLMs) and foundation models, introduce new privacy risks that extend beyond those observed in conventional deep learning systems. Due to their scale, extensive pretraining on heterogeneous data sources, and widespread deployment via public APIs, large models present a substantially expanded attack surface from a penetration testing perspective [69].

Privacy attacks against large models commonly exploit memorization and generative capabilities to extract sensitive training data. Empirical studies have demonstrated that carefully crafted prompts can induce models to reveal personally identifiable information, proprietary text, or verbatim training samples, constituting a form of training data extraction. Membership inference attacks have also been adapted to large models, where attackers assess whether specific data records were included in pretraining or fine-tuning corpora based on output confidence, response patterns, or likelihood scores [70]. Additionally, prompt injection and jailbreaking techniques can be leveraged to bypass safety constraints, enabling unauthorized access to sensitive model behaviors or hidden system prompts [71].

From a penetration testing [72] perspective, these attacks align primarily with reconnaissance and exploitation phases, where testers probe exposed model interfaces, analyze response variability, and systematically craft adversarial prompts to induce privacy leakage. Post-exploitation analysis focuses on quantifying the severity of disclosed information, assessing regulatory impact, and evaluating the robustness of deployed safeguards.

Defences against large-model privacy attacks include differential privacy—based training and fine-tuning, output filtering and redaction mechanisms, prompt sanitization, rate limiting, and continuous monitoring of anomalous query patterns. While these techniques can reduce privacy risk, they introduce trade-offs in utility, accuracy, and deployment [73]. Penetration testing plays a critical role in evaluating the effectiveness of these defences by simulating realistic adversarial behaviors and measuring residual leakage under controlled attack conditions.

Incorporating large-model privacy attacks into penetration testing frameworks is therefore essential for modern AI deployments. Unlike traditional security assessments, PT for large models must account for prompt-based interaction, probabilistic output behavior, and lifecycle-specific risks spanning pretraining, fine-tuning, and inference [74]. This highlights the need for adaptive, AI-aware penetration testing methodologies that extend beyond conventional network and application security paradigms.

4 Penetration Testing for DL-Related Privacy

As a safety assessment to identify weaknesses in a system by replicating real-world attacks [75], penetration testing for DL models requires to evaluate the models' defenses against possible adversarial threats, privacy violations, and ethical attacks. With AI-driven systems getting more integrated into cybersecurity, it is necessary to have DL-specific penetration testing for the purpose of analysing the security and privacy of DL systems.

4.1 Penetration Testing Approaches to DL Models

This section presents the main categories of penetration testing methods for DL models.

4.1.1 White-Box and Black-Box Testing

In **white-box testing,** testers have full access to DL models' structure, parameters, and training data. While white-box penetration testing is highly efficient at detecting internal flaws in model layouts, it might not accurately represent real-world attack scenarios where attackers do not have such direct access [34,76].

In **black-box testing,** testers are not given any prior information about the parameters, training data, or structure of the model. Testers engage with the system's API or external user interface and examine its output. Although being more realistic and useful, black-box testing may not be as good at identifying underlying architectural flaws as white-box methods [34,76].

Potential solution: If black-box and white-box testing can be merged, then it can restrict unauthorized access and model checks. Additionally, active learning strategies can reduce the number of inquiries, which is required for adversarial security assessment and model extraction.

4.1.2 Dynamic and Static Testing

Dynamic Penetration Testing refers to analyzing actual attack scenarios along with examining DL models in real time [77]. This type of testing includes the detection of abnormalities or unauthorized access and adversarial attacks during runtime to check a system's security strength. It also continuously monitors model flow and safety standards. Since dynamic testing offers real-time security assurance, it necessitates automatic assessment tools and continuous monitoring.

Static Penetration Testing encompasses the examination of a system's architecture, training data, and source code [77] without deploying the actual code. In this penetration testing, code analysis is used to identify a DL model's architectural flaw, with data scanning detecting possible data poisoning and adversarial pre-deployment training attempting to boost resilience. Static testing does not take real-time adversarial threats into consideration, even though it contributes to the early detection of possible vulnerabilities.

4.1.3 Automated Security Testing

Automated tools have been created to analyse model security and privacy issues due to the complexity of AI-related penetration testing.

IBM's Adversarial Robustness Toolbox (ART) is an open-source AI penetration testing tool, allowing researchers and developers to evaluate, authorize, and verify machine learning models [78]. ART also uses various defense approaches to assess privacy risks of membership inference atatcks.

Google TensorFlow is one of the most widely used penetration testing frameworks for DL models. It ensures regulatory compliance is followed, impairs the model's ability to remember private information, and reduces the risk of membership inference attacks [79].

Although automated AI penetration testing methods are essential for expanding security tests, their capacity to manage unpredictable adversarial threats remains limited.

4.2 Key Phases of Penetration Testing for DL-Related Privacy

A penetration testing framework for DL-related privacy should take a methodical approach so that it can systematically assess DL models' security and detect flaws. The key phases of such a framework are illustrated in Fig. 4 and expounded below.

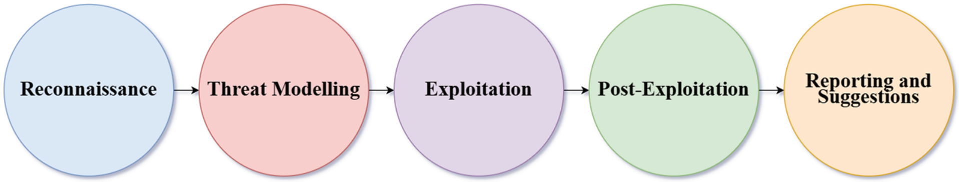

Figure 4: Penetration testing workflow for DL-related privacy.

(1) Reconnaissance. It is vital to gather comprehensive information regarding the target model, its structure, and any possible weaknesses prior to performing penetration testing [80–82]. This phase helps uncover any vulnerabilities that might be used to launch privacy attacks. This phase includes:

– Identifying the model type (e.g., transformer-based models, convolutional neural networks, and recurrent neural networks).

– Examining the characteristics of the dataset, and finding out if personally identifiable information, health records, or monetary transactions are present.

– Visualizing attack vectors, taking into account all endpoints, execution platforms, and public APIs.

– Analysing metadata to gather details about hyperparameters and training methods.

Note that unauthorized access to the model may be made possible by public APIs, opening the door for malicious requests. Model inversion and membership inference could be implemented easily using publicly accessible models.

Although the workflow in Fig. 4 follows a classical penetration testing structure, it is explicitly adapted for deep learning systems by incorporating model-centric assessment activities. Reconnaissance includes inference API discovery, model architecture identification, and dataset sensitivity analysis. Threat modeling focuses on DL-specific attacks such as model inversion, membership inference, extraction, and poisoning. Exploitation evaluates these attacks against training and inference pipelines, while post-exploitation quantifies privacy leakage, model integrity degradation, and regulatory impact. This adaptation ensures that the workflow reflects DL-specific attack surfaces rather than traditional network security testing.

(2) Threat Modeling. This phase pays particular attention to privacy issues, such as model exploitation, data leaks, and malicious attacks, which are likely to put DL models at risk [80,83]. This phase includes:

– Identifying attacks and ascertaining whether the attack is from inside, outside or fraudulent.

– Detecting the threat type (e.g., Adversarial Attacks, Model Inversion Attacks, Membership Inference Attacks, Model Extraction Attacks, or Data Poisoning Attacks).

– Assessing risks through investigating the probability and seriousness of any attack.

The absence of strong privacy-preserving measures makes DL models vulnerable to model inversion and membership inference attacks. Also, adversaries may use model APIs to extract decision-making parameters, which could result in model theft.

(3) Exploitation. In this phase, the model is influenced to manage attacks so as to determine its shortcomings [80,84]. This phase measures the model's strength to handle genuine privacy risks. This phase includes:

– Implementing active attacks (e.g., adversarial attacks, membership inference attacks, model inversion attacks, and data poisoning attacks) to evaluate the model's robustness and weaknesses.

– Assessing each attack, the possibility of success and impact on security and privacy.

Active attacks might weaken model performance, leading to inaccurate decision-making.

(4) Post-Exploitation. This phase determines how serious privacy leaks are and how well privacy-preserving methods work. It also provides information about the model's vulnerability to abuses including data restoration and prediction [80,84,85].

(5) Reporting and Suggestions. This final stage is to record the outcome of penetration testing and offer practical suggestions, making sure privacy-related rules are followed [84,86]. By gathering thorough reports on penetration testing, testers can measure security weaknesses, attack incidence results and the impact of mitigation strategies.

Based on the reports, safety standards with legal requirements and best industry practices can be investigated, and further recommendations for enhancing privacy protection can be made. Inadequate reporting and neglecting to implement recommended privacy protection measures could result in DL models to ongoing risks and regulatory violations [42,87–89].

4.3 Privacy-Preserving Methods

While differential privacy, homomorphic encryption, and federated learning are commonly described as privacy-preserving techniques, the tools implementing these methods also function as AI/ML-specific security mechanisms when evaluated through a penetration testing lens. In contrast to adversarial testing libraries that actively simulate attacks (e.g., model inversion or membership inference), DP, HE, and FL-based tools support defensive evaluation by enabling testers to measure privacy leakage, robustness under adversarial conditions, and compliance-related risk throughout the deep learning life-cycle. In this review, these tools are therefore treated as complementary AI/ML-specific security tools that extend penetration testing beyond attack execution to systematic assessment of privacy protection effectiveness. Since sensitive data is frequently handled by DL systems, privacy protection has become a priority. Many privacy-preserving schemes have been proposed in recent years to reduce the risks of membership inference attacks, model inversion attacks, and data leaks [90,91]. These schemes aim to preserve DL models' efficiency and performance while securing sensitive data and protecting privacy. Based on key factors such as accessibility, tool availability, and known areas of attack, Table 5 shows the suitability of the most well-known privacy-preserving techniques for penetration testing.

Table 5: Suitability of privacy-preserving techniques for penetration testing. More circles (•) represent a greater level of suitability.

Privacy-Preserving Technique	Penetration Testing Suitability
Differential Privacy	••••
Homomorphic Encryption	•••
Federated Learning	•••••

These privacy-preserving techniques are discussed below and an overview of these techniques and associated tools for deep learning systems are listed in Table 6.

Table 6: Mapping of privacy-preserving techniques and associated tools to penetration testing lifecycle phases for deep learning systems.

Techniques	Tool	PT Phases	DL Component	Evaluation Objective	Testing Capability
DP	TensorFlow Privacy	Exploitation/Post-Exploitation	Training pipeline	Measure privacy leakage under noise	High
DP	Opacus	Exploitation/Post-Exploitation	Training pipeline	Audit DP guarantees	High
HE	Microsoft SEAL	Exploitation	Inference pipeline	Protect data during computation	Medium
FL	PySyft	Reconnaissance /Exploitation	Distributed training	Detect leakage via updates	High
FL	FATE	Exploitation/Post-Exploitation	Federated aggregation	Secure model updates	High
ML Security	IBM ART	Exploitation	Inference API	Simulate privacy attacks	High

4.3.1 DP (Differential Privacy)

A mathematical framework called DP adds controllable noise (distraction or disturbance) to datasets or model outcomes to provide robust privacy assurances [90,92,93]. Penetration testing is an organized process for identifying security weaknesses, whereas DP is a statistical protection technique against such flaws [94,95]. Thus, penetration testing and DP can be interrelated for the purpose of privacy protection, especially in DL systems that handle private and/or sensitive data. Overall, DP can serve as both a barrier against privacy violations and a tool for assessing system flaws during penetration testing.

According to [90], "Penetration testing techniques, such as adversarial attack simulations and differential privacy analysis, play a crucial role in identifying vulnerabilities but frequently do not provide full coverage across all stages of a deep learning system's lifecycle". DP helps to restrict the impact of each specific data point by inserting controllable noise into the data or model outputs, lowering the danger of privacy breach when models are under attack [90,96]. On the one hand, penetration testing frequently assesses the accuracy of DP solutions, verifying that the added noise is enough to prevent inference attacks while preserving the model's efficiency. Penetration testing methods also guide the setting of DP parameters, striking a balance between privacy and performance. On the other hand, DP influences the direction and purpose of penetration testing, suggesting what needs additional examination because of security concerns.

– Examining the effectiveness of DP and addressing the issue of performance decline if needed.

– Figuring out how well the added noise hides confidential information to ensure that DP offers the required privacy protection.

– Confirming that the implementation of DP satisfies government regulations (e.g., GDPR, and HIPAA).

In summary, penetration testing is useful for verifying that DP meets the appropriate equilibrium of privacy, performance and usability for DL models.

Tools for DP From a penetration testing perspective, DP tools enable quantitative evaluation of privacy leakage and adversarial resilience under controlled attack scenarios. The acceptance of DP has resulted in the creation of several powerful tools that make it easier to apply DP in DL systems. These tools are particularly useful for DL models, where privacy of training data used is vital. This study covers four popular DP tools: TensorFlow Privacy, OpenDP, Opacus, and DiffPrivLib.

TensorFlow Privacy is an open-source library created by Google that enhances TensorFlow to allow for differentially private stochastic gradient descent or DP-SGD, a key algorithm that adds controlled noise to gradients during model training to ensure that each data point has minimal impact on the overall model, while offering basic privacy assurances [79]. Additionally, TensorFlow Privacy helps developers achieve DP on a broad scale, for example in production-grade DL scenarios.

OpenDP is an open-source project proposed by Harvard University in partnership with Microsoft to assist data analysts and organizations that require high data privacy without compromising performance, utilizing DP as a flexible and extendable framework for privacy-preserving data analysis [64]. OpenDP can securely evaluate confidential information by including precisely regulated noise while protecting privacy. Additionally, OpenDP is a tool that allows for unique privacy-oriented data analysis using modular blocks, while being cautious about privacy-induced cost.

Opacus is Facebook's DP library for machine learning services. It contains basic and special features [64,79] to serve both machine learning operators and skilled DP researchers. Its main objective is to ensure the privacy of each trainee's specimens while reducing any negative impact on the accuracy of completed models. Opacus also offers privacy management capabilities and privacy budgeting, suitable for medical, financial, and other area involving confidential information.

DiffPrivLib created by IBM for machine learning applications, provides DP variations of popular scikit-learn algorithms [64,79]. It includes a collection of tools for machine learning and data analytics, along with built-in privacy measures. Furthermore, DiffPrivLib is concerned with accessibility and obedience to legislative frameworks (e.g., GDPR) entailing robust privacy protections in data handling.

4.3.2 HE (Homomorphic Encryption)

HE is a cryptographic operation, enabling calculations to be made on encrypted information without decrypting it [97]. It can be applied to DL systems to secure data in estimation and training, assuring that confidential data remains hidden.

Penetration testing and HE are connected, because HE offers a strong cryptographic method to tackle detected hazards, especially those related to disclosure of information during data processing, while penetration testing detects such security loopholes [90]. Penetration testing techniques for DL systems are to identify DL models' secret parameters or expose confidential information from the data used for learning. After these flaws are identified, HE can be applied as a strong protective approach to preserve information without disclosing original text inputs on the server, while analysis is in progress.

HE guarantees that both input and output data remain encrypted throughout computation, thereby preventing access to sensitive information during processing [90]. By confirming no information leaks throughout the model's learning or during updates, penetration testing may verify whether HE is appropriately included within the model's timeline. Moreover, when assessing the capacity and effectiveness of DL models, penetration testing should take into account the computational expense that HE imposes. The coordination between penetration testing and HE is important for the security and privacy protection of DL systems.

The purpose of penetration testing for HE-based DL systems is to inspect the privacy protection of the HE scheme and its effect on computional cost. The appropriate testing areas are:

– Evaluating the encryption algorithm's durability during cryptographic attacks, and checking if the encryption might be compromised and if pen testers can address the harm.

– Analyzing how much computational complexity is added by HE.

– Verifying that the encryption algorithm can generate accurate results and HE works well with state-of-the-art DL frameworks (e.g., TensorFlow and PyTorch).

Tools for HE: Within penetration testing workflows, HE tools support assessment of confidentiality guarantees during inference and data processing under adversarial observation. Several HE libraries and tools have been developed to facilitate its application in practical scenarios. Some of the most widely used HE methods are described below:

Microsoft SEAL (Simple Encrypted Arithmetic Library) is an open source c++ library for HE, developed by Microsoft Research, which supports the CKKS (Brakerski/Fan-Vercauteren) scheme for approximate floating-point computations and the BFV (Brakerski-Gentry-Vaikuntanathan) scheme for exact integer arithmetic [98]. It is appropriate for programs that protect privacy and secure services in the cloud because of its efficient and flexible structure. It is widely adopted in both academic research and practical applications due to its performance and adaptability.

PALISADE is a lattice cryptography library written in C++ that implements a variety of HE techniques, such as BFV, BGV, CKKS, FHEW, and TFHE, developed by Dharpa [99]. PALISADE was designed with a modular architecture, making it suitable for both research and production contexts, especially in industry and government circumstances where powerful encryption features are needed. PALISADE is an effective tool for advanced cryptography research because of its wide scheme assistance, which enables developers to test out various HE techniques.

HElib (Homomorphic-Encryption Library) is an open-source C++ library developed by IBM, which focuses on the BGV and CKKS schemes, with particular emphasis on bootstrapping, which is a technique that enables an unlimited number of calculations on encrypted data [100]. When extensive control over fully homomorphic encryption (FHE) is required in educational and industry research, HElib is frequently utilized due to its abstraction of higher levels and the enhancement for complex processes. Additionally, its strengths allow it to be an ideal option for sophisticated FHE tests and implementations, yet it can have a longer learning process than SEAL or TenSEAL.

As a next-generation successor to PALISADE, **OpenFHE (Open Fully Homomorphic Encryption)** is an open-source C++ library for FHE that offers enhanced flexibility, modularity, and efficiency [97]. It supports a wide range of FHE schemes, such as BFV, BGV, and CKKS for arithmetic operations, as well as FHEW and TFHE. In addition to its modular design, OpenFHE allows researchers to seamlessly incorporate new cryptographic methods and improvements, while maintaining backward compatibility with PALISADE to simplify migration.

4.3.3 FL (Federated Learning)

In Federated Learning (FL), data is not stored on a single central server but remains on personal devices such as smartphones and IoT devices to train deep learning models [90,101,102]. Instead of sharing raw data, only the update models are sent to a central server for accumulation in federated learning. Since the raw information never leaves local devices, this method reduces the risk of data disclosure [36,95,103,104].

FL, which enables model training across decentralized data sources without transferring raw data to a central location, has emerged as an effective approach for privacy-preserving deep learning [105–108]. In delicate industries like banking, healthcare, where regulatory frameworks like HIPAA and GDPR demand strong data protection and compliance, this strategy is especially beneficial for them [109,110]. But it introduces new privacy and security vulnerabilities because of its distributed and shared structure. This makes penetration testing critical for assessing and improving the resilience of federated learning systems.

Penetration testing in FL is necessary to assess vulnerabilities such as gradient leakage, client or server manipulation, poisoning attacks, membership inference, and model inversion attacks. These attacks can either make the model less accurate or steal private information by exploiting the shared data (e.g., model updates or gradients) exchanged during training. As an instance, a malicious actor pretending to be a legitimate user in the FL system might attempt to infer private user data from the updates or inject poisoned gradients to corrupt the global model.

Incorporating penetration testing into FL helps to actively assess the effectiveness of privacy-enhancing technologies such as HE, secure multi-party computation (SMPC), secure aggregation, and differential privacy. However, the goal of these methods is to hide or secure private information, their implementations may still contain vulnerabilities [35,111]. On the other hand, penetration testing aid in verifying whether hostile tactics can still compromise security and whether the deployed protections offer enough security against realistic attack models. So, penetration testing acts as a safeguard to ensure that FL continues to uphold privacy, even if there are attackers trying to break it.

The objective of penetration testing in FL is to evaluate the privacy as well as security of data exchanges between the central server and participating devices [112,113]. The key testing areas include:

– Ensuring model updates are transmitted safely over encryption protocols, like TLS to protect against hacker eavesdropping.

– Inspecting the modified models if potential disclose of private data about the local information is made available. A penetration tester can assess whether attackers could reconstruct original data from observable model activity.

– Monitoring how resilient the system is to malicious clients that could potentially damage the model by submitting corrupted updates.

Through such assessments, penetration testing can simplify the identification of weaknesses in FL systems and guarantees the preservation of privacy during the collaborative training process [38].

Tools for FL: FL frameworks provide an evaluation surface for penetration testing by enabling analysis of leakage, poisoning, and manipulation risks in distributed training environments. A number of frameworks and tools have been created to simplify the implementation of FL. These frameworks make it easier to train models across distributed devices while preserving privacy and keeping data secure [114]. An overview of key tools and their features is provided below:

TensorFlow Federated (TFF) framework, developed by Google, is a python-based library to implement FL algorithms using TensorFlow [115]. It offers high-level APIs for testing and simulating FL ideas, making it particularly useful for research. Moreover, TFF simplifies experimentation by simulating multiple users with their own datasets [116]. It allows users to add privacy features like differential privacy and includes widely used training methods such as Federated Averaging (FedAvg) to protect data. Moreover, it works seamlessly with TensorFlow models, which makes it easy to reuse existing machine learning architectures in a privacy friendly environment.

PySyft developed by Openminded, is a python-based library that integrates FL with techniques like differential privacy, homomorphic encryption and secure multiparty computation (SMPC) to enable privacy-preserving machine learning. This framework integrates well with well-known machine learning frameworks such as TensorFlow and PyTorch [115]. It helps to preserve user confidentiality by enabling models to be trained on distributed data. Additionally, it offers confidential estimation without disclosing private information.

Flower is a flexible, framework-agnostic FL library that works well for both large-scale testing and real-world deployments. It is compatible with a variety of machine learning frameworks, including PyTorch, TensorFlow, and Scikit-learn [115]. Flower supports deployment across diverse devices and allows developers to design custom aggregation strategies to aggregate updates from various users.

FATE (Federated AI Technology Enabler), an industrial-grade federated learning platform developed by WeBank AI, is a high-quality FL platform that enables secure data processing and model training across multiple organizations. FATE supports different types of FL, including horizontal, vertical, and transfer learning, depending on the data distribution. It protects training data with robust privacy techniques including secure multi-party computing (SMPC) and differential privacy (DP) [115]. Furthermore, FATE supports a variety of machine learning models, such as neural networks, logistic regression, and decision trees.

OpenFL (Open Federated Learning) is a tool made by Intel, that helps different organizations work together to train machine learning models without sharing their raw data [117]. OpenFL employs a decentralized method in which data remains local, thereby enhancing privacy. It is effortless to setup in practical environments because it enables container-based deployments. Moreover, OpenFL prioritizes robust security of information by utilizing protected locations known as enclaves and secure computing to protect data during processing.

Table 6 maps privacy-preserving techniques and associated tools to standard penetration testing life-cycle phases (reconnaissance, exploitation, and post-exploitation) following PTES and NIST methodologies. The table highlights the deep learning components under evaluation and the primary testing objectives for assessing privacy leakage, model robustness, and regulatory impact.

By integrating DP, HE, and FL within a penetration testing—oriented evaluation framework, the proposed hybrid approach addresses the individual limitations of existing tools by jointly mitigating privacy leakage, robustness under adversarial testing, and deployment-level constraints across the deep learning life-cycle.

4.4 An Illustrative Case Study: Privacy-Oriented Penetration Testing of a DL Model

To demonstrate the practical applicability of the proposed penetration testing framework, we present an illustrative case study involving a deep learning image classification model deployed via a public inference API in a healthcare-like setting [118]. The model is assumed to be trained on sensitive medical imaging data and made accessible to external users for prediction services [119]. For clarity and brevity, this illustrative case study focuses on the core penetration testing phases where privacy leakage is manifested—reconnaissance, exploitation, and post-exploitation, while threat modeling and reporting are implicitly incorporated through attack selection and evaluation metrics.

Reconnaissance Phase: The penetration testing process begins by identifying exposed model interfaces, available output information (e.g., confidence scores), and access constraints. Public API documentation and response formats are analyzed to determine potential privacy leakage vectors [120].

Exploitation Phase: During exploitation, adversarial techniques such as membership inference and model inversion attacks are executed using DL-specific security tools (e.g., IBM Adversarial Robustness Toolbox). These attack classes are well-established mechanisms for inferring whether specific data records were included in the training set or for reconstructing sensitive features of the training data from model outputs [37]. Attack success is evaluated using metrics including membership inference accuracy, reconstruction similarity, and attack success rate.

Post-Exploitation Phase: Post-exploitation analysis assesses the severity of privacy leakage by quantifying the amount of sensitive information inferred and evaluating potential regulatory impact under data protection frameworks. Defensive mechanisms, such as differential privacy–based training and output filtering, are then evaluated by re-running attacks and measuring reductions in leakage metrics and corresponding impacts on model accuracy [34].

This illustrative case study demonstrates how the proposed framework extends beyond a conceptual workflow by enabling structured, lifecycle-based assessment of privacy risks and defenses under realistic attack scenarios. The case study is intended to showcase practical applicability rather than provide exhaustive empirical benchmarking.

5 Research Questions

Research questions serve as the foundational framework that guides systematic investigation and provides structured pathways for addressing complex challenges in deep learning privacy and security. In the context of this comprehensive review, research questions are particularly important because they help synthesize the diverse findings from the literature analysis, identify critical knowledge gaps that require immediate attention, and establish clear directions for future research endeavors. The formulation of targeted research questions enables researchers and practitioners to focus their efforts on the most pressing issues while providing measurable objectives for evaluating the effectiveness of proposed solutions. Furthermore, well-structured research questions facilitate the translation of theoretical insights into practical implementations that can address real-world privacy and security challenges in deep learning deployments.

The following research questions have been systematically derived based on the literature review findings and represent the most critical areas where current knowledge and practice fall short of addressing the evolving privacy and security landscape in deep learning systems.

Q1. What limitations do existing privacy-preserving mechanisms exhibit when evaluated through penetration testing across the deep learning lifecycle?

Penetration testing reveals that existing privacy-preserving mechanisms provide partial and phase-dependent protection across the deep learning life-cycle [36,120,121]. Differential privacy is effective at reducing inference-based leakage during training but remains vulnerable to model extraction and poisoning attacks at inference time. Homomorphic encryption protects data during computation but does not prevent privacy leakage through exposed model outputs or APIs. Federated learning limits raw data sharing; however, penetration testing of aggregation and update phases exposes susceptibility to gradient leakage, membership inference, and poisoning attacks. These findings demonstrate that individual privacy mechanisms do not provide comprehensive protection when assessed under adversarial testing conditions, underscoring the need for lifecycle-aware evaluation.

Q2. How can penetration testing be used to evaluate and balance the trade-off between privacy protection and model accuracy in deep learning systems under realistic attack scenarios?

Penetration testing enables quantitative evaluation of the privacy–accuracy trade-off by subjecting models to realistic adversarial conditions. Testing shows that stronger privacy controls, such as increased noise in differential privacy, reduce attack success rates but introduce measurable accuracy degradation [122]. Conversely, configurations optimized for performance exhibit higher vulnerability to inference and extraction attacks. By systematically varying privacy parameters and measuring outcomes such as attack success probability, privacy leakage magnitude, and accuracy loss, penetration testing supports evidence-based calibration of privacy mechanisms that balances security requirements with operational utility.

Q3. Which deep learning–specific attack vectors are most effective at different penetration testing phases, and what system-level factors contribute to their success?

Penetration testing indicates that the effectiveness of deep learning–specific attacks varies across testing phases and system configurations. Model inversion and membership inference attacks are most effective during exploitation and post-exploitation phases, particularly in systems exposing probabilistic outputs via public APIs [67,120]. Model extraction attacks primarily succeed during iterative exploitation, where unrestricted querying enables functional replication. Data poisoning attacks are most impactful during training and update phases, especially in distributed or federated learning environments lacking robust validation. Attack success is strongly influenced by model architecture, API exposure, training pipeline design, and data governance assumptions, highlighting the importance of system-level analysis in penetration testing.

6 Challenges and Future Directions

6.1 Challenges

The intersection of deep learning technologies with cybersecurity presents unprecedented challenges that fundamentally diverge from traditional security paradigms. As DL systems become increasingly integrated into critical infrastructure and sensitive applications, the security community faces a complex landscape of emerging vulnerabilities that conventional defensive strategies cannot adequately address. These challenges are compounded by the rapid pace of AI advancement, which often outpaces the development of corresponding security measures, creating persistent gaps in protection capabilities. The following analysis examines the primary obstacles that currently impede effective security assessment and privacy protection in deep learning environments, each representing a critical area where current approaches fall short of addressing the unique requirements of AI system security.

1. Inefficiencies of Traditional Penetration Testing Techniques. The unique vulnerabilities inherent in deep learning models, such as data poisoning, membership inference, and model inversion attacks, are not adequately addressed by traditional cybersecurity frameworks (e.g., OWASP, PTES, NIST, and OSSTMM), which were originally designed for conventional software systems and network infrastructures [32]. These established frameworks lack the specialized methodologies and assessment criteria required to evaluate AI-specific attack vectors that target model training processes, inference mechanisms, and data handling procedures. Consequently, these frameworks often fail to provide a comprehensive evaluation of deep learning system security, leaving critical vulnerabilities undetected and organizations exposed to sophisticated AI-targeted attacks.

2. Deficiency in Implementing Privacy Methods. Even though numerous privacy-preserving strategies have been proposed and validated in theoretical contexts, significant constraints such as excessive computational expenses, system complexity, and integration challenges with practical production environments have substantially complicated their real-world implementation [123]. The gap between theoretical privacy guarantees and practical deployment capabilities remains substantial, with many organizations struggling to balance privacy requirements with operational efficiency and system performance. Furthermore, the lack of robust, evidence-based validation studies and standardized implementation guidelines hampers wider adoption and undermines organizational confidence in these methods, creating a cycle where promising privacy technologies remain underutilized despite their theoretical benefits.

3. Continuous Transformation of Attacks. Deep learning (DL) systems are increasingly vulnerable to powerful and dynamic threats that extend beyond traditional threat models and evolve at an unprecedented pace [124]. The adversarial landscape in AI security is characterized by rapid innovation in attack methodologies, where defensive measures quickly become obsolete as attackers develop more sophisticated techniques. Therefore, most existing security tools remain reactive in nature, typically responding only after a deep learning system has been compromised and showing fundamental limitations in proactive threat detection and prevention capabilities. They lack the predictive capabilities required to identify and prevent new, advanced attacks, particularly those that exploit the unique characteristics of machine learning algorithms and training processes.

4. Lack of Tools and Technologies. Current privacy-preserving toolkits including PySyft, TensorFlow Privacy, and ART provide only minimal support for comprehensively securing deep learning systems in production environments [123]. These tools, while valuable for research and proof-of-concept implementations, often fail to provide essential enterprise features such as scalability, interoperability, and seamless integration with diverse deep learning model architectures and deployment platforms. As a result, achieving continuous security monitoring and comprehensive testing remains a significant challenge for organizations, particularly those operating large-scale AI systems that require robust, automated security assessment capabilities.

5. Regulatory Compliance. DL systems face considerable challenges in complying with stringent regulatory standards like GDPR and HIPAA, which impose strict requirements for data protection, user privacy, and system accountability. For implementing effective technical security measures that ensure privacy protection, DL systems must support complex services such as continuous user activity monitoring, comprehensive action logging, and guaranteed compliance with evolving legal requirements [125]. However, integrating these compliance capabilities into existing DL architectures and workflows is frequently complex and difficult to achieve in practice, particularly when organizations must balance compliance requirements with model performance, operational efficiency, and development timelines.

6.2 Future Directions

This literature review utilizes a heterogeneous research methodology that includes a thorough examination of prior research, observational testing, and an evaluation of existing frameworks to address ongoing privacy concerns in deep learning models and to investigate the trade-off between privacy and accuracy. This research explores why current penetration testing methods prove insufficient, examines how privacy mechanisms influence model accuracy, and evaluates the persistent effectiveness of adversarial attacks against modern defense systems.

Based on systematic evaluation of scholarly sources, this paper identifies privacy-enhancing strategies such as differential privacy (DP), homomorphic encryption (HE), and federated learning (FL), alongside penetration testing strategies specifically tailored for deep learning (DL) systems. To advance the practical security of DL systems and address critical gaps in current privacy protection approaches, future research must tackle several fundamental challenges that currently impede the effective implementation of existing privacy evaluation techniques:

1. Comprehensive Real-World Privacy Solution Implementation. While penetration testing techniques are actively being researched in relation to DL systems, their practical adoption in operational environments remains substantially limited and inadequately validated. As cyberattacks become increasingly sophisticated and targeted, dependence solely on theoretical recommendations proves insufficient without rigorous real-world implementation and empirical validation studies. Furthermore, vulnerabilities that are unique to DL architectures, such as data poisoning, membership inference, and model inversion attacks, are not adequately addressed by current cybersecurity frameworks, including OWASP, PTES, NIST, and

OSSTMM. Therefore, future research should prioritize the development and deployment of these theoretical concepts in practical operational contexts to achieve demonstrable and measurable privacy protection outcomes [4,44,54].

2. Advanced Hybrid Privacy Technique Integration and Optimization. A sophisticated hybrid approach that strategically combines DP and FL represents a promising methodology for enhancing data privacy while maintaining acceptable model performance levels. Specifically, DP introduces carefully calibrated computational noise into training data to protect individual information and prevent adversaries from reconstructing sensitive data, while FL enables decentralized model training without requiring raw data sharing [95]. However, to fully realize the transformative potential of this hybrid approach, extensive empirical studies are essential to evaluate its effectiveness across diverse application domains and under varying adversarial threat models, including adaptive attacks that specifically target hybrid defense mechanisms.

3. Optimized Privacy-Performance Trade-Off Resolution. Model accuracy degradation caused by privacy-enhancing methods presents a particularly critical challenge in high-stakes domains such as healthcare diagnostics and autonomous vehicle systems, where even minimal performance reductions can result in catastrophic consequences. Future research must therefore focus on developing innovative strategies that provide robust privacy assurances while maintaining optimal utility levels through advanced techniques such as selective privacy application, dynamic privacy budgets, and context-aware privacy mechanisms.

4. Development of Automated DL Privacy Testing Frameworks. Although existing tools such as ART, TensorFlow Privacy, and PySyft establish a foundational basis for privacy testing in DL environments, their automation capabilities, scalability features, and comprehensive coverage of potential attack vectors remain significantly insufficient. Consequently, there exists an urgent need for developing sophisticated, automated testing frameworks specifically engineered for DL systems that enable continuous evaluation, seamless integration into production machine learning pipelines, and comprehensive attack surface coverage.

5. Enhanced Regulatory Compliance and Legal Accountability Framework Development. Regulatory compliance represents a fundamental requirement for modern DL deployments, and future frameworks must be explicitly architected to satisfy stringent regulations such as GDPR and HIPAA through integration of comprehensive risk assessment capabilities, detailed activity logging mechanisms, and complete traceability features to ensure data security and legal accountability. The absence of these critical capabilities exposes organizations to substantial reputational risks and legal liabilities [125].

6. Implementation of Adaptive and Proactive Defense Mechanisms. Current DL model security approaches are predominantly reactive in nature, focusing on threat mitigation only after vulnerabilities have been exploited by adversaries. To address this fundamental limitation, future research should integrate machine learning-based anomaly detection systems, prioritize advanced threat intelligence and predictive analysis capabilities, and establish secure automated update pipelines that can proactively respond to emerging threats before they can be exploited.

As threat landscapes continue to evolve with increasing sophistication and persistence, there exists a critical imperative to continuously advance and refine privacy testing methodologies. Successfully addressing these multifaceted challenges is essential for developing DL systems that demonstrate robustness, transparency, and effective privacy protection capabilities in real-world, adversarial operational environments.

7 Discussion and Conclusion

This comprehensive review has systematically examined the fundamental limitations of traditional penetration testing frameworks when confronted with the unique privacy vulnerabilities and security challenges inherent in deep learning (DL) systems. The investigation reveals that established cybersecurity standards such as OWASP, PTES, NIST, and OSSTMM, despite their proven effectiveness in detecting conventional network and system vulnerabilities, fundamentally lack the specialized capabilities and methodological approaches required to assess DL-specific threat vectors. These conventional frameworks are inadequately equipped to address sophisticated attacks including adversarial manipulations, data poisoning campaigns, model inversion techniques, and membership inference exploits that specifically target machine learning architectures and training processes.

The analysis conducted throughout this survey demonstrates that the cybersecurity landscape has evolved beyond the scope of traditional security assessment methodologies, particularly as DL systems become increasingly prevalent in sensitive domains such as healthcare, finance, and autonomous systems. The unique attack surface presented by machine learning models introduces novel privacy risks that conventional penetration testing tools cannot adequately evaluate or mitigate. The survey further illuminates the complex landscape of privacy-preserving techniques, revealing that while methods such as homomorphic encryption (HE), differential privacy (DP), and federated learning (FL) offer promising protection mechanisms for sensitive data processing, they remain constrained by significant practical limitations. These limitations include substantial computational overheads that impact system performance, inherent accuracy-privacy trade-offs that compromise model utility, and the relentless evolution of adversarial strategies that continuously challenge existing defensive measures. The dynamic nature of adversarial attack development means that static privacy protection mechanisms often become obsolete as attackers develop more sophisticated techniques to circumvent established defenses. Consequently, no single privacy-preserving solution currently provides comprehensive, adaptive, or sustained protection for DL models in real-world operational deployments, creating critical security gaps that require immediate attention.

To address these fundamental gaps in current security assessment capabilities, this paper emphasizes the urgent need for developing dedicated penetration testing frameworks specifically tailored to evaluate and enhance DL privacy protection mechanisms. Such specialized frameworks must integrate multiple complementary approaches including structured adversarial testing protocols, automated vulnerability assessment procedures, and regulatory compliance verification processes, thereby ensuring comprehensive robustness against realistic attack scenarios that DL systems face in production environments. The proposed framework architecture should incorporate continuous monitoring capabilities, adaptive threat modeling, and dynamic response mechanisms that can evolve alongside emerging attack vectors. Furthermore, hybrid protection strategies, such as the strategic combination of FL with DP techniques, present a promising pathway toward achieving optimal balance between strong privacy guarantees and minimal accuracy degradation while maintaining practical deployability.

In summary, this comprehensive review makes significant contributions to the field by (i) systematically identifying and categorizing critical DL privacy vulnerabilities that are consistently overlooked by traditional security testing approaches, (ii) conducting thorough assessment of the strengths, limitations, and practical applicability of current privacy-preserving methods and associated toolkits, and (iii) proposing a novel hybrid, automated, and regulation-compliant penetration testing paradigm specifically designed for comprehensive DL system security evaluation. The proposed framework contribute to more proactive, adaptive, and specialized security assessment methodologies that can effectively address the unique challenges posed by machine learning deployments.

Acknowledgement: None.

Funding Statement: The work of Yanrong Lu in this paper was supported in part by the Tianjin Natural Science Foundation Project (24JCZDJC01000), and the Fundamental Research Funds for the Central Universities of China (No. 3122025091).

Author Contributions: Conceptualization, Salma Akther, Wencheng Yang; methodology, Salma Akther, Wencheng Yang; writing—original draft preparation, Salma Akther, Shicheng Wei, Wencheng Yang; writing—review and editing, Wencheng Yang, Song Wang, Ji Zhang, Xu Yang, Yanrong Lu, and Yan Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hayat T, Gatlin K. AI-powered ethical hacking: rethinking cyber security penetration testing. ResearchGate. 2025. doi:10.13140/RG.2.2.18954.38080.
2. Cookey IB, Ajoku CM, Ogini PB. The impact of ethical hacking on identifying security vulnerabilities: analyzing the effectiveness of evolving penetration testing methodologies. Int J Institutional Leadersh Policy Manag. 2025;7(2):202–11. doi:10.21275/sr24608143458.
3. Li M, Zhu T, Yan H, Chen T, Lv M. HER-PT: an intelligent penetration testing framework with Hindsight Experience Replay. Comput Secur. 2025;152:104357. doi:10.2139/ssrn.4932007.
4. Alhamed M, Rahman MMH. A Systematic literature review on penetration testing in networks: future research directions. Appl Sci. 2023;13(12):6986. doi:10.3390/app13126986.
5. Maji S, Jain H, Pandey V, Siddiqui VA. White hat security-an overview of penetration testing tools. In: Proceedings of the Advancement in Electronics & Communication Engineering. Bandung, Indonesia: IAES; 2022. p. 58–65.
6. Vats P, Mandot M, Gosain A. A comprehensive literature review of penetration testing & its applications. In: 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Piscataway, NJ, USA: IEEE; 2020. p. 674–80.
7. Kozlovska M, Piskozub A, Khoma V. Artificial intelligence in penetration testing: leveraging AI for advanced vulnerability detection and exploitation. Artif Intell. 2025;10(1):67–72. doi:10.23939/acps2025.01.065.
8. Alencar RC, Fernandes BJT, Lima PHES, Da Silva CMR. AI techniques for automated penetration testing in MQTT networks: a literature investigation. Int J Comput Appl. 2025;47(1):106–21. doi:10.1080/1206212x.2024.2443504.
9. Koroniotis N, Moustafa N, Turnbull B, Schiliro F, Gauravaram P, Janicke H. A deep learning-based penetration testing framework for vulnerability identification in internet of things environments. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Piscataway, NJ, USA: IEEE; 2021. p. 887–94.
10. Tariq MI, Memon NA, Ahmed S, Tayyaba S, Mushtaq MT, Mian NA, et al. A review of deep learning security and privacy defensive techniques. Mob Inf Syst. 2020;2020:1–18. doi:10.1155/2020/6535834.
11. Yaqoob I, Hussain SA, Mamoon S, Naseer N, Akram J. Penetration testing and vulnerability assessment. J Netw Commun Emerg Technol. 2017;7(8):10–8.
12. Wang H, Shang D, Jin Z, Liu F. A multi-graph combination screening strategy enabled graph convolutional network for alzheimer's disease diagnosis. IEEE Trans Instrum Meas. 2024;73:4012319. doi:10.1109/tim.2024.3485439.
13. Wang H, Li Z, Han X, Zhang G, Zhang Q, Zhang D, et al. MAG-Net: a multiscale adaptive generation network for PET synthetic CT. IEEE Trans Rad Plasma Med Sci. 2024;9(1):83–94. doi:10.1109/trpms.2024.3418831.
14. Liu F, Liang SN, Jaward MH, Ong HF, Wang H, Initiative ADN, et al. CRAD: cognitive aware feature refinement with missing modalities for early alzheimer's progression prediction. Comput Med Imaging Graph. 2025;126(1):102664. doi:10.1016/j.compmedimag.2025.102664.
15. Yang F, Wang H, Wei S, Sun G, Chen Y, Tao L. Multi-model adaptive fusion-based graph network for Alzheimer's disease prediction. Comput Biol Med. 2023;153(16):106518. doi:10.1016/j.compbiomed.2022.106518.
16. Cheng J, Wang H, Wei S, Mei J, Liu F, Zhang G. Alzheimer's disease prediction algorithm based on de-correlation constraint and multi-modal feature interaction. Comput Biol Med. 2024;170(3):108000. doi:10.1016/j.compbiomed.2024.108000.

17. Wei S, Yang W, Wang E, Wang S, Li Y. A 3D decoupling Alzheimer's disease prediction network based on structural MRI. Health Inf Sci Syst. 2025;13(1):17. doi:10.1007/s13755-024-00333-3.

18. Nguyen TPV, Yang W, Tang Z, Xia X, Mullens AB, Dean JA, et al. Lightweight federated learning for STIs/HIV prediction. Sci Rep. 2024;14(1):6560. doi:10.1038/s41598-024-56115-0.

19. Suleski T, Ahmed M, Yang W, Wang E. A review of multi-factor authentication in the Internet of Healthcare Things. Digit Health. 2023;9:20552076231177144. doi:10.1177/20552076231177144.

20. Tang Z, Van Nguyen TP, Yang W, Xia X, Chen H, Mullens AB, et al. High security and privacy protection model for STI/HIV risk prediction. Digit Health. 2024;10:20552076241298425. doi:10.1177/20552076241298425.

21. Chen Y, Zhao C, Xu Y, Nie C, Zhang Y. Deep learning in financial fraud detection: innovations, challenges, and applications. Data Sci Manag. 2025;36:44. doi:10.1016/j.dsm.2025.08.002.

22. Mahmoudi Y. Algorithmic trading and price forecasting: machine learning and deep learning strategies for financial markets. Qual Quant. 2025;59(5):4757–78. doi:10.1007/s11135-025-02179-7.

23. Tang C, Abbatematteo B, Hu J, Chandra R, Martín-Martín R, Stone P. Deep reinforcement learning for robotics: a survey of real-world successes. In: Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto, CA, USA: AAAI Press; 2025. p. 28694–8.

24. Djerbi R, Rouane A, Taleb Z, Safia S. Design and implementation of a self-driving car using deep reinforcement learning: a comprehensive study. Comput Ind Eng. 2025;207(6):111319. doi:10.1016/j.cie.2025.111319.

25. Yang W, Wang S, Cui H, Tang Z, Li Y. A review of homomorphic encryption for privacy-preserving biometrics. Sensors. 2023;23(7):3566. doi:10.3390/s23073566.

26. Yang W, Wang S, Hu J, Tao X, Li Y. Feature extraction and learning approaches for cancellable biometrics: a survey. CAAI Trans Intell Technol. 2024;9(1):4–25. doi:10.1049/cit2.12283.

27. Yang W, Wang S, Wu D, Tang Z, Yang X, Cui H, et al. Enhancing privacy in face recognition with dual-path feature compression and homomorphic encryption. In: IEEE International Joint Conference on Biometrics (IJCB). Piscataway, NJ, USA: IEEE; 2025.

28. Wang H, Cheng Y, Zheng W, Cheng J, Li X, Li M, et al. A multi-illumination dataset and an illumination domain adaptation network for finger vein identification. In: Proceedings of the 33rd ACM International Conference on Multimedia. New York, NY, USA: ACM; 2025. p. 7940–8.

29. Wang H, Ruan J, Fan C, Cheng Y, Lv Z. ID-RemovalNet: identity removal network for EEG privacy protection with enhancing decoding tasks. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence; 2025 Aug 16–22; Montreal, QC, Canada. p. 4209–17.

30. Wang H, Wang M, Liu X, Cheng Y, Liu F, Zhou J, et al. The cancelable multimodal template protection algorithm based on random index. IEEE Trans Emerg Top Comput. 2025;13(3):1200–14. doi:10.1109/tetc.2025.3574359.

31. Alguliyev RM, Aliguliyev RM, Abdullayeva FJ. Privacy-preserving deep learning algorithm for big personal data analysis. J Ind Inf Integr. 2019;15(1):1–14. doi:10.1016/j.jii.2019.07.002.

32. Shukla R, Kumar CRS. A structured approach to simulating penetration testing in cyber ranges. TechRxiv. 2025. doi:10.36227/techrxiv.174362706.68995054/v1.

33. Alwabisi SO. AI in penetration testing: a systematic mapping study. TechRxiv. 2025. doi:10.36227/techrxiv.175099664.46246512/v1.

34. Zhang G, Liu B, Zhu T, Zhou A, Zhou W. Visual privacy attacks and defenses in deep learning: a survey. Artif Intell Rev. 2022;55(6):4347–4401. doi:10.1007/s10462-021-10123-y.

35. Thapa C, Arachchige PCM, Camtepe S, Sun L. Splitfed: when federated learning meets split learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto, CA, USA: AAAI Press; 2022. p. 8485–93.

36. Chen J, Yan H, Liu Z, Zhang M, Xiong H, Yu S. When federated learning meets privacy-preserving computation. ACM Comput Surv. 2024;56:319:1–36. doi:10.1145/3679013.

37. Nasr M, Shokri R, Houmansadr A. Machine learning with membership privacy using adversarial regularization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2018. p. 634–46. doi:10.1145/3243734.3243855.

38. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Found Trends® Mach Learn. 2021;14(1–2):1–210. doi:10.1561/2200000083.
39. Liu X, Xie L, Wang Y, Zou J, Xiong J, Ying Z, et al. Privacy and security issues in deep learning: a survey. IEEE Access. 2021;9:4566–93. doi:10.1109/access.2020.3045078.
40. Isnaini K, Asyari MH, Amrillah SF, Suhartono D. Vulnerability assessment and penetration testing on student service center system. ILKOM J Ilmi. 2024;16(2):161–71. doi:10.33096/ilkom.v16i2.1969.161-171.
41. Hung NV, Cong NT. Applying reinforcement learning in automated penetration testing. J Sci Technol Inf Secur. 2022;3(17):61–77. doi:10.1109/cns48642.2020.9162301.
42. Alhogail A, Alkahtani M. Automated extension-based penetration testing for web vulnerabilities. Procedia Comput Sci. 2024;238:15–23. doi:10.1016/j.procs.2024.05.191.
43. Li Z, Zhang Q, Yang G. EPPTA: efficient partially observable reinforcement learning agent for penetration testing applications. Eng Rep. 2025;7(1):e12818. doi:10.22541/au.169406476.64066230/v1.
44. Shebli HMZA, Beheshti BD. A study on penetration testing process and tools. In: 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). Piscataway, NJ, USA: IEEE; 2018. p. 1–7.
45. Schiller T, Caulkins B, Wu AS, Mondesire S. Security awareness in smart homes and internet of things networks through swarm-based cybersecurity penetration testing. Information. 2023;14(10):536. doi:10.3390/info14100536.
46. Yang Y, Chen L, Liu S, Wang L, Fu H, Liu X, et al. Behaviour-diverse automatic penetration testing: a coverage-based deep reinforcement learning approach. Front Comput Sci. 2025;19(3):193309. doi:10.1007/s11704-024-3380-1.
47. Shah S, Mehtre BM. A modern approach to cyber security analysis using vulnerability assessment and penetration testing. Int J Elect Commun Comput Eng. 2013;4:47–52. doi:10.1109/icaccct.2014.7019182.
48. Zhang X, Shen W, Wang Y, Cui L, Liang Z. Data security assessment method based on penetration testing. In: 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC). Piscataway, NJ, USA: IEEE; 2024. p. 1–6.
49. Ziro A, Gnatyuk S, Toibayeva S. Improved method for penetration testing of web applications. In: IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security; 2023 Mar 22–24; Khmelnytskyi, Ukraine. p. 518–28.
50. Filiol E, Mercaldo F, Santone A. A method for automatic penetration testing and mitigation: a red hat approach. Procedia Comput Sci. 2021;192(1):2039–46. doi:10.1016/j.procs.2021.08.210.
51. Sinchana K, Sinchana C, Gururaj HL, Kumar BS. Performance evaluation and analysis of various network security tools. In: 2019 International Conference on Communication and Electronics Systems (ICCES). Piscataway, NJ, USA: IEEE; 2019. p. 644–50.
52. Goyal P, Goyal A. Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark. In: 2017 9th International Conference on Computational Intelligence and Communication Networks (CICN). Piscataway, NJ, USA: IEEE; 2017. p. 77–81.
53. Mabsali NA, Jassim H, Mani J. Effectiveness of wireshark tool for detecting attacks and vulnerabilities in network traffic. In: 1st International Conference on Innovation in Information Technology and Business (ICIITB 2022). AL Den Haag, The Netherlands: Atlantis Press; 2023. p. 114–35.
54. Adam HM, Putra GD. A review of penetration testing frameworks, tools, and application areas. In: 2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE). Piscataway, NJ, USA: IEEE; 2023. p. 319–24.
55. Manjunath S, Malshetty S, Jayalakshmi D, Banger C, Vali YS. A comprehensive NIDS-based strategy for web application penetration testing. SSRG Int J Comput Sci Eng. 2024;11:1–6. doi:10.14445/23488387/ijcse-v11i12p101.
56. Yusuf MF, Hikmah IR, Amiruddin, Sunaringtyas SU. Security testing of XYZ website application using ISSAF and OWASP WSTG v4.2 Methods. Teknika. 2025;14(1):66–77. doi:10.34148/teknika.v14i1.1156.
57. DeCusatis C, Peko P, Irving J, Teache M, Laibach C, Hodge J. A framework for open source intelligence penetration testing of virtual health care systems. In: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). Piscataway, NJ, USA: IEEE; 2022. p. 0760–4.

58. Gunawan TS, Lim MK, Kartiwi M, Malik NA, Ismail N. Penetration testing using Kali linux: SQL injection, XSS, wordpres, and WPA2 attacks. Indones J Electr Eng Comput Sci. 2018;12(2):729–37. doi:10.11591/ijeecs.v12.i2.pp729-737.
59. Astrida DN, Saputra AR, Assaufi AI. Analysis and evaluation of wireless network security with the penetration testing execution standard (PTES). Sink J Dan Penelit Tek Inform. 2021;6(1):147–54. doi:10.33395/sinkron.v7i1.11249.
60. Wijaya I, Sasmita GMA, Pratama I. Web application penetration testing on Udayana University's OASE E-learning platform using information system security assessment framework (ISSAF) and open source security testing methodology manual (OSSTMM). Int J Inf Technol Comput Sci. 2024;16(2):45–56. doi:10.5815/ijitcs.2024.02.04.
61. Wang T, Bi Z, Zhang Y, Liu M, Hsieh W, Feng P, et al. Deep learning model security: threats and defenses. arXiv: 2412.08969. 2024.
62. Nicolae MI, Sinn M, Tran MN, Buesser B, Rawat A, Wistuba M, et al. Adversarial robustness toolbox v1.0.0. arXiv: 1807.01069. 2019.
63. Chakraborty A, Alam M, Dey V, Chattopadhyay A, Mukhopadhyay D. Adversarial attacks and defences: a survey. arXiv: 1810.00069. 2018.
64. Zhang S, Hagermalm A, Slavnic S, Schiller EM, Almgren M. Evaluation of open-source tools for differential privacy. Sensors. 2023;23(14):6509. doi:10.3390/s23146509.
65. OWASP Foundation. OWASP risk rating methodology [Internet]. 2021 [cited 2026 Jan 12]. Available from: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology.
66. Tabassi E. Artificial intelligence risk management framework (AI RMF 1.0) [Internet]. 2023 [cited 2026 Jan 12]. Available from: http://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf.
67. Yang W, Wang S, Wu D, Cai T, Zhu Y, Wei S, et al. Deep learning model inversion attacks and defenses: a comprehensive survey. Artif Intell Rev. 2025;58(8):242. doi:10.1007/s10462-025-11248-0.
68. Zhao J, Chen Y, Zhang W. Differential privacy preservation in deep learning: challenges, opportunities and solutions. IEEE Access. 2019;7:48901–48911. doi:10.1109/access.2019.2909559.
69. Li H, Chen Y, Luo J, Wang J, Peng H, Kang Y, et al. Privacy in large language models: attacks, defenses and future directions. arXiv:2310.10383. 2024.
70. Yi S, Liu Y, Sun Z, Cong T, He X, Song J, et al. Jailbreak attacks and defenses against large language models: a survey. arXiv:2407.04295. 2024.
71. Deng G, Liu Y, Robotics A, Klagenfurt AAU, Liu P, Li Y, et al. PentestGPt: evaluating and harnessing large language models for automated penetration testing. In: 33rd USENIX Security Symposium (USENIX Security 24). Philadelphia, PA, USA: USENIX Association; 2024. p. 847–64.
72. Happe A, Cito J. On the surprising efficacy of LLMs for penetration-testing. arXiv:2507.00829. 2025.
73. Das BC, Amini MH, Wu Y. Security and privacy challenges of large language models: a survey. ACM Comput Surv. 2025;57(6):1–39. doi:10.1145/3712001.
74. Wang S, Zhu T, Liu B, Ding M, Ye D, Zhou W, et al. Unique security and privacy threats of large language models: a comprehensive survey. ACM Comput Surv. 2025;58:83:1–36. doi:10.1145/3764113.
75. Ashawa M, Kpelai SJ. Penetration testing: analysis of emerging technologies and their impact on pen testing. Int J Eng Technol Inform. 2023;4(4):1–4. doi:10.51626/ijeti.2023.04.00064.
76. Bae H, Jang J, Jung D, Jang H, Ha H, Lee H, et al. Security and privacy issues in deep learning. arXiv:1807.11655. 2018.
77. Dencheva L. Comparative analysis of Static application security testing (SAST) and Dynamic application security testing (DAST) by using open-source web application penetration testing tools [master's thesis]. Dublin, Ireland: National College of Ireland; 2022.
78. Huang Y, Hu H, Chen C. Robustness of on-device models: adversarial attack to deep learning models on android apps. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). Piscataway, NJ, USA: IEEE; 2021. p. 101–10.

79. Said HE, Mahmoud QH, Goyal M, Hashim F. Comparative analysis of differential privacy implementations on synthetic data. In: 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC). Piscataway, NJ, USA: IEEE; 2025. p. 00243–9.

80. Ankele R, Marksteiner S, Nahrgang K, Vallant H. Requirements and recommendations for IoT/IIoT models to automate security assurance through threat modelling, security analysis and penetration testing. In: Proceedings of the 14th International Conference on Availability, Reliability and Security. New York, NY, USA: ACM; 2019. p. 1–8.

81. Chowdhary A, Huang D, Mahendran JS, Romo D, Deng Y, Sabur A. Autonomous security analysis and penetration testing. In: 2020 16th International Conference on Mobility, Sensing and Networking (MSN). Piscataway, NJ, USA: IEEE; 2020. p. 508–15.

82. Abdulsatar M, Ahmad H, Goel D, Ullah F. Towards deep learning enabled cybersecurity risk assessment for microservice architectures. Clust Comput. 2025;28(6):1–16. doi:10.1007/s10586-024-05092-0.

83. Kaushik S, El Madhoun N. Analysis of blockchain security: classic attacks, cybercrime and penetration testing. In: 2023 Eighth International Conference on Mobile and Secure Services (MobiSecServ). Piscataway, NJ, USA: IEEE; 2023. p. 1–6.

84. Nagendran K, Adithyan A, Chethana R, Camillus P, Varshini KBS. Web application penetration testing. Int J Innov Technol Explor Eng. 2019;8(10):1029–35. doi:10.35940/ijitee.j9173.0881019.

85. Moreno AC, Hernandez-Suarez A, Sanchez-Perez G, Toscano-Medina LK, Perez-Meana H, Portillo-Portillo J, et al. Analysis of autonomous penetration testing through reinforcement learning and recommender systems. Sensors. 2025;25(1):211. doi:10.3390/s25010211.

86. Aibekova A, Selvarajah V. Offensive security: study on penetration testing attacks, methods, and their types. In: 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). Piscataway, NJ, USA: IEEE; 2022. p. 1–9.

87. Mirjalili M, Nowroozi A, Alidoosti M. A survey on web penetration test. Adv Comput Sci Int J. 2014;3(6):107–21.

88. Jayasuryapal G, Pranay PM, Kaur H. A survey on network penetration testing. In: 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM). Piscataway, NJ, USA: IEEE; 2021. p. 373–8.

89. Chowdappa KB, Lakshmi SS, Kumar PP. Ethical hacking techniques with penetration testing. Int J Comput Sci Inf Technol. 2014;5(3):3389–93. doi:10.4018/978-1-6684-8218-6.ch012.

90. Vayghan BJ. Privacy of deep learning systems: a penetration testing framework. Authorea Prepr. 2024. doi:10.36227/techrxiv.172296728.87985494/v1.

91. Ali S, Irfan MM, Bomai A, Zhao C. Towards privacy-preserving deep learning: opportunities and challenges. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA). Piscataway, NJ, USA: IEEE; 2020. p. 673–82.

92. Alkurd R, Abualhaol I, Yanikomeroglu H. Preserving user privacy in personalized networks. IEEE Netw Lett. 2021;3(3):124–8. doi:10.1109/lnet.2021.3094518.

93. Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2015. p. 1310–21.

94. Kerschbaum F. Towards privacy in deep learning. In: 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA). Piscataway, NJ, USA: IEEE; 2021. p. 279–80.

95. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, et al. A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. New York, NY, USA: ACM; 2019. p. 1–11.

96. Gandhi VJ, Shokeen S, Koshti S. A systematic literature review on privacy of deep learning systems. arXiv:2212.04003. 2022.

97. Marcolla C, Sucasas V, Manzano M, Bassoli R, Fitzek FHP, Aaraj N. Survey on Fully homomorphic encryption, theory, and applications. Proc IEEE. 2022;110(10):1572–1609. doi:10.36227/techrxiv.19315202.

98. Dhiman S, Mahato GK, Chakraborty SK. Homomorphic encryption library, framework, toolkit and accelerator: a review. SN Comput Sci. 2023;5(1):24.

99. Al-Janabi AA, Al-Janabi STF, Al-Khateeb B. Secure data computation using deep learning and homomorphic encryption: a survey. Int J Online Biomed Eng. 2023;19(11):53. doi:10.3991/ijoe.v19i11.40267.
100. Doan TVT, Messai ML, Gavin G, Darmont J. A survey on implementations of homomorphic encryption schemes. J Supercomput. 2023;79(13):15098–15139. doi:10.1007/s11227-023-05233-z.
101. Abanilla S, Chatterjee M, Dass S. Towards privacy preserving financial fraud detection. In: 2023 International Conference on Machine Learning and Applications (ICMLA). Piscataway, NJ, USA: IEEE; 2023. p. 1723–6.
102. Duan M, Liu D, Chen X, Tan Y, Ren J, Qiao L, et al. Astraea: self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In: 2019 IEEE 37th International Conference on Computer Design (ICCD). Piscataway, NJ, USA: IEEE; 2019. p. 246–54.
103. Hasan MM. Federated learning models for privacy-preserving AI in enterprise decision systems. Int J Bus Econ Insights. 2025;5(3):238–69. doi:10.63125/ry033286.
104. Mohammadi S, Balador A, Sinaei S, Flammini F. Balancing privacy and performance in federated learning: a systematic literature review on methods and metrics. J Parallel Distrib Comput. 2024;192(7):104918. doi:10.1016/j.jpdc.2024.104918.
105. Aledhari M, Razzak R, Parizi RM, Saeed F. Federated learning: a survey on enabling technologies, protocols, and applications. IEEE Access. 2020;8:140699–725. doi:10.1109/access.2020.3013541.
106. Wen J, Zhang Z, Lan Y, Cui Z, Cai J, Zhang W. A survey on federated learning: challenges and applications. Int J Mach Learn Cybern. 2023;14(2):513–35. doi:10.1007/s13042-022-01647-y.
107. Banabilah S, Aloqaily M, Alsayed E, Malik N, Jararweh Y. Federated learning review: fundamentals, enabling technologies, and future applications. Inf Process Manag. 2022;59(6):103061. doi:10.1016/j.ipm.2022.103061.
108. Mehmood MH, Khan MI, Ibrahim A. Balancing privacy and accuracy: federated learning with differential privacy for medical image data. In: 2024 7th International Conference on Data Science and Information Technology (DSIT). Piscataway, NJ, USA: IEEE; 2024. p. 1–6.
109. Lyu L, Yu H, Yang Q. Threats to federated learning: a survey. arXiv:2003.02133. 2020.
110. Zhang C, Xie Y, Bai H, Yu B, Li W, Gao Y. A survey on federated learning. Knowl Based Syst. 2021;216(1):106775. doi:10.1016/j.knosys.2021.106775.
111. Neto HNC, Hribar J, Dusparic I, Mattos DMF, Fernandes NC. A survey on securing federated learning: analysis of applications, attacks, challenges, and trends. IEEE Access. 2023;11:41928–53. doi:10.1109/access.2023.3269980.
112. Yan Z, Li D, Zhang Z, He J. Accuracy—security tradeoff with balanced aggregation and artificial noise for wireless federated learning. IEEE Internet Things J. 2023;10(20):18154–67. doi:10.1109/jiot.2023.3277632.
113. Wang Z, Wang Z, Fan X, Wang C. Federated learning with domain shift eraser. In: Proceedings of the Computer Vision and Pattern Recognition Conference. Piscataway, NJ, USA: IEEE; 2025. p. 4978–87.
114. Khoa TV, Alsheikh MA, Alem Y, Hoang DT. Balancing security and accuracy: a novel federated learning approach for cyberattack detection in blockchain networks. arXiv:2409.04972. 2024.
115. Guendouzi BS, Ouchani S, El Assaad H, El Zaher M. A systematic review of federated learning: challenges, aggregation methods, and development tools. J Netw Comput Appl. 2023;220(8):103714. doi:10.1016/j.jnca.2023.103714.
116. Saidi R, Moulahi T, Aladhadh S, Zidi S. Advancing federated learning: optimizing model accuracy through privacy-conscious data sharing. In: 2024 IEEE 25th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). Piscataway, NJ, USA: IEEE; 2024. p. 64–9.
117. Reina GA, Gruzdev A, Foley P, Perepelkina O, Sharma M, Davidyuk I, et al. OpenFL: an open-source framework for Federated Learning. arXiv:2105.06413. 2021.
118. Shamshirband S, Fathi M, Dehzangi A, Chronopoulos AT, Alinejad-Rokny H. A review on deep learning approaches in healthcare systems: taxonomies, challenges, and open issues. J Biomed Inform. 2021;113:103627. doi:10.1016/j.jbi.2020.103627.
119. Zargar HH, Zargar SH, Mehri R. Review of deep learning in healthcare. arXiv: 2310.00727. 2023.
120. Rigaki M, Garcia S. A survey of privacy attacks in machine learning. ACM Comput Surv. 2023;56(4):1–34. doi:10.1145/3624010.

121. Niu J, Liu P, Zhu X, Shen K, Wang Y, Chi H, et al. A survey on membership inference attacks and defenses in machine learning. J Inf Intell. 2024;2(5):404–54. doi:10.1016/j.jiixd.2024.02.001.
122. Zhang X, Zhang Q. Defending against attacks in deep learning with differential privacy: a survey. Artif Intell Rev. 2025;58(11):347. doi:10.1007/s10462-025-11350-3.
123. Prabowo S, Putrada AG, Oktaviani ID, Abdurohman M, Janssen M, Nuha HH, et al. Privacy-preserving tools and technologies: government adoption and challenges. IEEE Access. 2025;13(5):33904–34. doi:10.1109/access.2025.3540878.
124. Gopinath D. Analyzing cybersecurity dynamics: unveiling evolving threat patterns and modern attack vectors. Kristu Jayanti J Comput Sci. 2024;4(1):44–53. doi:10.59176/kjcs.v4i1.2433.
125. Teichmann FM, Boticiu SR. An overview of the benefits, challenges, and legal aspects of penetration testing and red teaming. Int Cybersecur Law Rev. 2023;4(4):387–97. doi:10.1365/s43439-023-00100-2.