WCRG Global Research WORLD COMET RESEARCH GROUP

International Journal of Data Engineering and Intelligent Computing
Vol. 9, No. 3, 2026
Doi: 10.67228/30715717/IJDEIC-V9I3P101
PP. 1-32

Original Article

Data-Centric Security for Generative AI Systems: Protecting Sensitive Data against Leakage, Inference Attacks, and Unauthorized Model Access

Santosh Kumar Jadala

Cyber Security & Business Analysis Specialist, Independent Researcher, USA.

Received: 07-02-2026 Revised: 07-21-2026 Accepted: 07-30-2026 Published: 08-02-2026

ABSTRACT

The growing use of generative artificial intelligence in enterprise and public-sector environments has introduced significant concerns regarding the protection of sensitive data. Generative AI systems process large volumes of information across training, fine-tuning, retrieval, inference, and output stages, creating multiple opportunities for unauthorized disclosure and misuse. This study examines data-centric security as an approach for protecting sensitive information against data leakage, membership inference, model inversion, prompt-based attacks, retrieval-related exposure, model extraction, and unauthorized access to AI services. It reviews major attack surfaces across the generative AI lifecycle and evaluates security controls including data classification, minimization, encryption, privacy-preserving learning, identity and access management, secure retrieval-augmented generation, output filtering, data loss prevention, and continuous monitoring. Based on these findings, the study proposes a data-centric security framework that links data sensitivity, access policies, model controls, and governance requirements throughout the AI lifecycle. The framework emphasizes that security controls should remain tied to sensitive information regardless of where the data is stored, processed, retrieved, or generated. The study provides practical guidance for organizations seeking to reduce privacy and confidentiality risks while maintaining controlled and accountable use of generative AI systems.

KEYWORDS

Generative Artificial Intelligence, Data-Centric Security, Data Leakage, Inference Attacks, Model Security, Sensitive Data Protection, Access Control, Privacy-Preserving AI.

ISSN: 3071-5717
PP: 1-32.

1. INTRODUCTION

Generative artificial intelligence has moved rapidly from experimental use into business operations, public services, education, healthcare, finance, software development, and knowledge management. Large language models and related generative systems are increasingly connected to organizational databases, document repositories, application programming interfaces, cloud services, and internal communication platforms. This expansion has improved the ability of organizations to automate information-intensive tasks, support decision-making, retrieve knowledge, and interact with large volumes of unstructured data. At the same time, it has introduced a distinct security problem: sensitive information is no longer confined to conventional databases and applications but may also appear in training datasets, fine-tuning records, embeddings, prompts, retrieved documents, model outputs, interaction histories, and model parameters.

This changing data environment challenges traditional approaches to information security. Conventional security architectures have largely focused on protecting networks, systems, applications, and storage environments through perimeter controls, authentication, encryption, firewalls, and endpoint security. These mechanisms remain essential, but they do not fully address the ways in which generative models process, reproduce, infer, and expose information. A user may have legitimate access to a model while still using carefully designed prompts to extract information that should not be disclosed. Similarly, a model may reveal information learned from its training data even when the underlying dataset is no longer directly accessible. Carlini et al. (2021) demonstrated that large language models can reproduce identifiable sequences from their training data, showing that model access can become an indirect route to sensitive information. This finding raises concerns about treating trained models as independent from the data used to create them.

1.1. Background and Motivation

The security implications of generative AI are closely connected to the scale and diversity of data required for model development and operation. Large models are commonly trained on extensive datasets and may later be adapted through fine-tuning, retrieval-augmented generation, or enterprise-specific knowledge sources. In organizational settings, these processes may involve customer information, employee records, financial documents, proprietary research, software code, legal materials, trade secrets, credentials, and other confidential data. Once such information enters an AI workflow, several copies or representations may exist across data stores, vector databases, temporary contexts, logs, caches, model checkpoints, and generated responses.

Research on language-model privacy has shown that personal and confidential information can be exposed through several mechanisms. Lukas et al. (2023), for example, examined the leakage of personally identifiable information from language models and demonstrated that privacy risks can persist when sensitive information is embedded within model behavior. Memorization is particularly important because a model may reproduce rare or distinctive training examples when prompted under suitable conditions. Carlini et al. (2023) further showed that memorization varies across neural

ISSN: 3071-5717
PP: 1-32.

language models and is influenced by factors such as model scale, data duplication, and characteristics of the training material.

The introduction of retrieval-augmented generation has created another important layer of risk. RAG systems allow models to retrieve external documents at inference time rather than relying solely on information contained in their parameters. This architecture can improve factual relevance and provide access to current or organization-specific knowledge, but it also connects the model directly to potentially sensitive information repositories. If retrieval permissions are poorly implemented, a user may receive information from documents that they would not normally be permitted to access. Zeng et al. (2024) found that retrieval-augmented generation introduces privacy concerns associated with the interaction between language models and external knowledge sources. Consequently, protecting only the underlying database is insufficient if the retrieval system can deliver restricted information to an authorized model session without enforcing the user's original access permissions.

These developments provide the motivation for a data-centric approach to generative AI security. Rather than concentrating primarily on the location of information or the infrastructure that stores it, data-centric security focuses on the sensitivity, ownership, permitted use, and movement of the data itself. Protection decisions can therefore remain connected to information as it moves through collection, preprocessing, training, storage, retrieval, inference, output generation, and logging. This approach is particularly relevant to generative systems because the same sensitive information may pass through several technical components during a single interaction.

1.2. Emerging Data-Security Risks in Generative AI

The threat landscape surrounding generative AI extends beyond accidental disclosure. Sensitive information may be targeted through training-data extraction, membership inference, model inversion, attribute inference, prompt injection, indirect prompt injection, model extraction, credential compromise, and unauthorized access to AI services.

Membership inference attacks seek to determine whether a particular record or example was included in the data used to train a model. Such attacks create privacy concerns even when the original record itself is not directly reconstructed. Shokri et al. (2017) established the practical significance of membership inference attacks against machine-learning models, while later research has examined how similar risks apply to modern language models. Model inversion presents a related concern because an adversary may use model responses or confidence information to reconstruct sensitive characteristics associated with training data. Fredrikson et al. (2015) demonstrated that information revealed through model interfaces can support the reconstruction of sensitive attributes under certain conditions.

Prompt-based attacks further complicate the security problem because users interact with generative systems through natural language rather than predetermined software commands. Malicious instructions may attempt to override application rules, reveal hidden instructions, access

ISSN: 3071-5717
PP: 1-32.

restricted context, or manipulate connected tools. Greshake et al. (2023) demonstrated that indirect prompt injection can compromise applications that integrate language models with external content and services. The risk becomes more serious when a model is connected to corporate databases, email systems, document repositories, browsing tools, or autonomous actions.

Unauthorized model access also creates threats to both sensitive data and intellectual property. Attackers may compromise API credentials, misuse privileged accounts, exploit weak authorization controls, or repeatedly query a model to recover information about its structure or behavior. Tramèr et al. (2016) showed that machine-learning models can be extracted through prediction APIs, establishing model theft as a practical security concern. More recently, Carlini et al. (2024) demonstrated that significant information about a production language model can be recovered through carefully designed interactions, highlighting the continuing relevance of model extraction in commercial generative systems.

Together, these risks show that confidentiality cannot be protected solely by securing the model endpoint. Security controls must account for who is accessing the system, what information is being processed, where that information originated, what the model is permitted to retrieve, what it is allowed to disclose, and how abnormal interactions are identified.

1.3. Research Problem

Despite growing research on language-model security and privacy, many existing protections remain separated across different technical and organizational areas. Privacy-preserving learning focuses on limiting information leakage from training data. Identity and access management governs users and service accounts. Data loss prevention examines sensitive information leaving controlled environments. Application security addresses malicious inputs and interface vulnerabilities, while model security considers extraction, manipulation, and unauthorized use. These areas address important parts of the problem, but treating them independently can leave gaps across the generative AI lifecycle.

For example, an organization may encrypt its vector database and apply strong authentication to its AI application, yet still expose confidential documents if retrieval authorization does not respect document-level permissions. A model may also be securely hosted while memorized training information remains recoverable through repeated queries. Similarly, a user with valid credentials may cause unauthorized disclosure through prompt manipulation if output controls and contextual authorization are weak. Such cases demonstrate that technical access to a model and legitimate access to the information processed by that model are not necessarily the same thing.

The central research problem is therefore the absence of a sufficiently integrated security approach that follows sensitive data across the full lifecycle of a generative AI system. Protection is required at collection, preprocessing, training, fine-tuning, model storage, retrieval, prompt

ISSN: 3071-5717
PP: 1-32.

processing, inference, output generation, logging, and monitoring stages. Each stage introduces different risks, yet security decisions need to remain connected to common information-classification and access policies.

Addressing this problem requires a framework that combines data discovery and classification, data minimization, encryption, privacy-preserving training, identity and access management, secure retrieval, prompt controls, model and API protection, output filtering, data loss prevention, monitoring, and governance. Such an approach can provide organizations with a clearer basis for determining how sensitive information should be handled before, during, and after interaction with generative AI systems.

1.4. Research Aim

The aim of this study is to develop a data-centric security framework for protecting sensitive information throughout the generative AI lifecycle against data leakage, inference attacks, and unauthorized model access.

The study seeks to examine how sensitive data is exposed across model training, fine-tuning, retrieval, inference, and output processes and to determine how existing security and privacy controls can be integrated into a coherent protection strategy. Particular attention is given to training-data leakage, membership inference, model inversion, prompt-based disclosure, retrieval-related exposure, model extraction, and unauthorized access to models and AI services.

The proposed framework places the sensitivity and permitted use of information at the center of security decision-making. It considers controls such as data classification, minimization, encryption, privacy-preserving learning, fine-grained authorization, secure retrieval, output filtering, data loss prevention, model-access controls, and continuous monitoring. By connecting these controls across the generative AI lifecycle, the study aims to provide both a conceptual foundation for further research and practical guidance for organizations seeking to deploy generative AI while maintaining appropriate protection of confidential, personal, and proprietary information.

2. LITERATURE REVIEW AND THEORETICAL FOUNDATIONS

Generative artificial intelligence systems depend on large volumes of data during training, fine-tuning, retrieval, inference, and post-processing. This dependence has made data protection a central security concern. Unlike conventional information systems, where sensitive information is mainly stored in databases, files, and applications, generative systems can retain, reproduce, infer, retrieve, and transform information through several interconnected components. These components include training datasets, model parameters, prompts, vector embeddings, external knowledge stores, system instructions, tool interfaces, logs, and generated outputs. As a result, security controls must address not only where information is stored but also how it is represented, accessed, processed, and disclosed throughout the system lifecycle.

ISSN: 3071-5717
PP: 1-32.

2.1. Generative AI Systems and the Data Lifecycle

A typical generative AI environment involves several stages through which data may pass. These include data collection, preparation, model training, fine-tuning, storage, deployment, prompt processing, retrieval, inference, output generation, and logging. Each stage introduces different security risks. During data preparation and training, organizations may expose personal, proprietary, financial, or operational information to systems that can later reproduce parts of that information. During deployment, prompts may contain confidential material that becomes part of temporary context, logs, or downstream processing. Retrieval-based applications introduce further exposure by connecting models to enterprise document repositories and vector databases.

Research has shown that language models can memorize portions of their training data and reproduce them under appropriate prompting conditions. Carlini et al. (2021) demonstrated that training examples could be extracted from large language models, including text that contained potentially sensitive information. Later work by Carlini et al. (2023) showed that memorization is influenced by factors such as duplication, model size, and characteristics of the training data. These findings indicate that information does not cease to require protection simply because it has been incorporated into model training. The trained model itself may become a secondary location from which information can be recovered.

The data lifecycle therefore extends beyond traditional storage boundaries. Sensitive information may exist as raw data, tokenized sequences, embeddings, model representations, prompt context, retrieved passages, generated responses, and audit logs. Security models that focus only on files and databases may overlook these alternative forms of exposure.

2.2. From Perimeter Security to Data-Centric Security

Traditional cybersecurity has relied heavily on protecting networks, endpoints, applications, and infrastructure boundaries. Firewalls, intrusion detection systems, identity management, encryption, and endpoint protection remain important, but generative AI introduces circumstances in which an authorized user may still cause unauthorized disclosure. A user may have legitimate access to an AI system while lacking permission to view the sensitive information that the model retrieves or generates.

Data-centric security addresses this problem by making the sensitivity and permitted use of information the basis for security decisions. Instead of assuming that access to an application implies access to all information handled by that application, data-centric controls evaluate the classification, ownership, purpose, and access requirements attached to individual data assets. This approach is closely related to the principles of least privilege and Zero Trust, where access is granted according to verified identity, context, resource sensitivity, and operational need.

ISSN: 3071-5717
PP: 1-32.

Within generative AI, this principle means that a model should not automatically receive unrestricted access to all available information. The user, model, retrieval system, tool, and external service should each operate within defined permissions. Data classification can further determine whether information may be used for training, fine-tuning, retrieval, external processing, logging, or output generation. The theoretical basis of data-centric security therefore rests on continuous control over data rather than reliance on the security of a single application or network boundary.

2.3. Sensitive Data in Generative AI

Generative AI systems may process personally identifiable information, financial records, health information, employee data, customer records, credentials, intellectual property, proprietary code, internal business documents, legal records, and security configurations. These categories vary in sensitivity and regulatory importance, but all can create serious consequences if disclosed without authorization.

The risk is particularly important because sensitive information may enter generative systems indirectly. Employees may include confidential data in prompts, upload internal documents to AI tools, or connect generative applications to enterprise knowledge repositories. Once processed, information may be retained in logs, used for later context, stored in agent memory, or exposed through generated responses. Lukas et al. (2023) demonstrated that language models can leak personally identifiable information, reinforcing concerns about sensitive information embedded within model behavior.

Agent-based systems introduce additional complications because conversational memory and persistent storage can retain information across sessions. Wang et al. (2025) examined privacy risks associated with language-model agent memory and showed that memory mechanisms can become an additional point of sensitive information exposure. This extends the data-security problem beyond the underlying language model to the wider system architecture in which the model operates.

2.4. Data Memorization and Training-Data Leakage

Memorization refers to the ability of a model to retain specific examples from its training data rather than learning only generalized statistical patterns. Memorization becomes a security issue when a model reproduces confidential or personal information after receiving carefully designed inputs.

Carlini et al. (2019) demonstrated unintended memorization in neural networks, while Carlini et al. (2021) later showed that language models could reveal actual training examples through extraction attacks. The risk is especially significant for rare or duplicated records because these examples may be easier for models to memorize. Kandpal et al. (2022) found that deduplicating training data can reduce privacy risks, indicating that data preparation practices can materially influence downstream exposure.

ISSN: 3071-5717
PP: 1-32.

Training-data leakage therefore cannot be addressed solely through access restrictions placed on the original dataset. Controls must also consider the behavior of the trained model. Privacy testing, deduplication, minimization, differential privacy, and memorization assessment can reduce the likelihood that sensitive examples are retained in recoverable form.

2.5. Inference Attacks

Inference attacks seek to obtain information that is not directly provided through the normal model interface. These attacks are particularly relevant to generative systems because repeated interaction can reveal details about training data, user records, or model behavior.

Membership inference attacks attempt to determine whether a particular record was used during training. Shokri et al. (2017) demonstrated that machine-learning models can reveal information about training-set membership through their outputs. Nasr et al. (2019) later extended privacy analysis to stronger attack settings and showed that models may reveal considerably more information when attackers have greater access.

Model inversion attacks focus on reconstructing sensitive features from model outputs. Fredrikson et al. (2015) demonstrated that confidence information could be exploited to recover sensitive attributes associated with training records. Attribute inference represents a related threat in which an attacker derives private characteristics from information available through the model.

Training-data extraction is another form of inference-based disclosure. In this case, the attacker aims to recover memorized sequences directly. Recent research has shown that even aligned production systems may remain susceptible to extraction under sufficiently persistent querying. Nasr et al. (2025) demonstrated scalable extraction methods against production language models, showing that alignment and deployment safeguards do not necessarily eliminate training-data exposure.

2.6. Prompt-Based Data Leakage

Prompt-based attacks exploit the instruction-following behavior of generative models. Attackers may attempt to override system instructions, reveal hidden prompts, bypass safety rules, access restricted context, or cause connected tools to perform unauthorized operations.

Direct prompt injection occurs when a user explicitly provides malicious instructions. Indirect prompt injection is more difficult to control because hostile instructions may be embedded in external content retrieved by the model. Greshake et al. (2023) demonstrated that indirect prompt injection can compromise applications that connect language models to external data and services. In such cases, the model may interpret attacker-controlled content as trusted instructions rather than as untrusted data.

Prompt-based leakage is particularly serious in enterprise systems because generative models often operate with access to internal documents, email, databases, search tools, and automated

ISSN: 3071-5717
PP: 1-32.

workflows. A successful prompt attack can therefore become a path to unauthorized data disclosure. Effective defenses require clear separation between instructions and retrieved content, context isolation, prompt filtering, tool permission controls, and output inspection.

2.7. RAG and Vector Database Security

Retrieval-augmented generation has become a common method for connecting language models to external knowledge. In a RAG system, a user query is converted into an embedding and matched against indexed documents. Relevant passages are then supplied to the language model as context for generating a response.

This architecture improves access to current and organization-specific information but introduces new security concerns. Sensitive documents may be indexed without proper classification, and retrieval systems may return information without applying the original document permissions. Vector databases may also contain representations derived from confidential data, while retrieved documents can carry malicious instructions or poisoned content.

Zeng et al. (2024) showed that RAG systems present privacy risks associated with the interaction between retrieval mechanisms and language models. Huang et al. (2023) similarly examined the privacy implications of retrieval-based language models and found that external knowledge retrieval can create additional information exposure. In addition, Zou et al. (2025) demonstrated that poisoned documents can manipulate RAG outputs by influencing what information is retrieved and presented to the model.

A secure RAG system therefore requires more than accurate retrieval. It must enforce document-level authorization, verify the provenance of indexed data, separate tenants, monitor retrieval behavior, and restrict the model from receiving information that the requesting user is not entitled to access.

2.8. Unauthorized Model Access and Model Theft

Generative AI systems can also be targeted through unauthorized access to models, APIs, repositories, or deployment infrastructure. Attackers may obtain credentials, compromise API keys, exploit overly broad permissions, or misuse legitimate accounts. Once access is gained, the attacker may seek to extract sensitive outputs, steal model capabilities, reconstruct model behavior, or gain access to connected enterprise data.

Model extraction has been recognized as a security concern for many years. Tramèr et al. (2016) demonstrated that prediction APIs can be used to replicate machine-learning models through repeated querying. More recent work has shown that similar risks remain relevant for advanced language models. Carlini et al. (2024) demonstrated that substantial information about a production language

ISSN: 3071-5717
PP: 1-32.

model could be recovered through carefully designed queries, illustrating that model interfaces may reveal aspects of protected intellectual property.

Protecting model access therefore requires strong identity controls, least-privilege permissions, secrets management, rate limiting, behavioral monitoring, API gateways, and controls on model repositories. Security monitoring should also distinguish legitimate high-volume use from extraction attempts or unusual access patterns.

Overall, the literature indicates that generative AI data security cannot be treated as a single technical problem. Leakage, inference, prompt manipulation, retrieval abuse, and unauthorized model access originate from different parts of the system, yet each ultimately concerns control over sensitive information. This supports the need for an integrated data-centric security framework that applies consistent protection principles throughout the generative AI lifecycle.

3. RESEARCH METHODOLOGY

3.1. Research Design

This study adopts a structured literature review and conceptual framework development approach. The methodology is designed to identify the principal data-security threats affecting generative AI systems, examine the controls proposed in previous studies, and organize these findings into a data-centric security framework.

A literature-based design is appropriate because research on generative AI security is distributed across several areas, including privacy, machine learning security, language-model security, access control, prompt injection, retrieval security, model extraction, and data governance. Reviewing these areas together allows the study to identify relationships that may not be visible when each threat is examined independently.

The study does not attempt to measure the effectiveness of a particular commercial platform or model. Instead, it develops a security architecture from peer-reviewed research and established security principles. The resulting framework is intended to provide a structured basis for analyzing sensitive-data protection across different generative AI implementations.

3.2. Literature Search Strategy

The literature search focuses on peer-reviewed publications addressing generative AI security, language-model privacy, machine-learning inference attacks, model extraction, retrieval security, prompt injection, and data-protection mechanisms. Relevant studies may be identified from established academic databases and publication repositories, including Scopus, Web of Science, IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and major machine-learning and natural-language-processing conference proceedings.

ISSN: 3071-5717
PP: 1-32.

Search terms are organized around the main threat categories addressed by the research. Example search combinations include:

- "generative AI" AND "data leakage"
- "large language model" AND "training data extraction"
- "LLM" AND "membership inference"
- "language model" AND "model inversion"
- "generative AI" AND "prompt injection"
- "retrieval augmented generation" AND "privacy"
- "RAG" AND "data security"
- "large language model" AND "model extraction"
- "generative AI" AND "access control"
- "LLM" AND "data loss prevention"

Backward and forward citation tracing may also be used to identify foundational and closely related studies. Foundational works on membership inference, model inversion, differential privacy, and model extraction are included where they provide the theoretical basis for more recent generative AI security research.

3.3. Inclusion and Exclusion Criteria

Studies are included when they meet at least one of the following conditions: they examine sensitive-data leakage from generative or machine-learning models; investigate membership inference, model inversion, attribute inference, or training-data extraction; analyze prompt injection or indirect prompt injection; address privacy and security in retrieval-augmented generation; examine model extraction or unauthorized model access; or propose controls relevant to data-centric protection.

Priority is given to peer-reviewed journal articles and papers published in established conference proceedings. Studies published up to 2025 are considered to maintain consistency with the scope of the research. Seminal earlier studies are retained when they establish attack models or security concepts that remain directly applicable to modern generative systems.

Studies are excluded when they focus solely on model performance, natural-language generation quality, general AI ethics without a clear security or privacy component, or cybersecurity topics that do not directly relate to generative AI data protection.

3.4. Analytical Dimensions

The selected literature is analyzed across several dimensions to support consistent comparison. First, each study is classified according to the primary asset at risk, such as training data, prompts, embeddings, retrieved documents, model parameters, API credentials, or generated outputs. Second, studies are categorized by attack type, including data leakage, membership inference, model inversion, prompt injection, retrieval manipulation, model extraction, or unauthorized access.

ISSN: 3071-5717
PP: 1-32.

Third, each threat is mapped to the stage of the generative AI lifecycle where exposure occurs. These stages include data collection, data preparation, training, fine-tuning, model storage, deployment, prompt processing, retrieval, inference, output generation, and logging. Fourth, proposed defenses are grouped according to their security function, including prevention, detection, containment, monitoring, and governance.

The analysis also considers the main security objective addressed by each control. These objectives include confidentiality, privacy preservation, integrity, controlled access, accountability, and model protection. Where available, limitations identified by the original authors are recorded, particularly when a defense introduces performance costs, reduces model utility, depends on restrictive assumptions, or addresses only a narrow attack scenario.

The final stage of the methodology synthesizes these findings into a data-centric security framework. Rather than treating security controls as isolated mechanisms, the framework connects them to the sensitivity and permitted use of information across the full generative AI lifecycle. This structure provides the basis for the threat model and security architecture developed in the following sections.

4. THREAT MODEL FOR DATA-CENTRIC GENERATIVE AI SECURITY

A threat model for data-centric generative AI security should identify what needs protection, who may attempt to compromise it, how attacks can occur, and where controls should be applied. Generative AI systems differ from conventional applications because sensitive information may exist in several forms at the same time. Data may appear as raw training records, fine-tuning examples, embeddings, retrieved documents, prompt context, model parameters, generated outputs, or system logs. Protecting only the original database therefore provides limited assurance once the same information has entered other parts of the AI environment.

The threat model adopted in this study treats sensitive data as the central security asset. It considers confidentiality, integrity, privacy, controlled access, and accountability throughout the generative AI lifecycle. Particular attention is given to training-data leakage, inference attacks, prompt manipulation, retrieval abuse, model extraction, credential compromise, and unauthorized access to connected enterprise information.

4.1. Protected Assets

The first stage of the threat model is the identification of assets that may contain, represent, or provide access to sensitive information. Training datasets are among the most important assets because they may include personally identifiable information, financial records, proprietary documents, software code, intellectual property, customer information, or internal communications. Fine-tuning datasets present similar risks, particularly when organizations customize general-purpose models with business-specific information.

ISSN: 3071-5717
PP: 1-32.

Model parameters must also be treated as protected assets. Although parameters do not function as traditional databases, research has shown that trained models may retain information about individual examples. Carlini et al. (2021) demonstrated that large language models can reproduce training sequences under targeted extraction conditions. This means that the model itself may become an indirect source of sensitive information.

Prompts and system instructions are another important category. User prompts may contain confidential business information, personal details, passwords, code, or other restricted material. System prompts may reveal application rules, tool configurations, security instructions, or internal architecture. If these prompts are stored in conversation history or operational logs, the exposure period can extend beyond the original session.

Embeddings and vector databases require similar protection. Retrieval-augmented generation systems create numerical representations of organizational documents and use them to identify relevant information during inference. Although embeddings are not direct copies of the source documents, the retrieval environment may still expose confidential information when access controls are weak. Zeng et al. (2024) showed that retrieval-augmented generation introduces privacy concerns because external knowledge can be exposed through model responses.

Other protected assets include model checkpoints, API keys, authentication tokens, agent memory, tool credentials, retrieved documents, generated outputs, conversation histories, audit logs, and model repositories. Each asset should be assigned a sensitivity classification and an explicit access policy.

4.2. Threat Actors

Threat actors may originate both outside and inside the organization. External attackers may attempt to obtain access through stolen credentials, compromised API tokens, prompt injection, automated querying, or vulnerabilities in supporting infrastructure. Their objectives may include extracting confidential data, replicating model behavior, stealing intellectual property, or gaining access to connected enterprise systems.

Malicious or curious users represent another important threat category. A user may have legitimate permission to access the AI application while deliberately attempting to retrieve information outside their authorized scope. This distinction is important because application access does not automatically imply permission to access every data source available to the model.

Privileged insiders may pose greater risks because they can access model repositories, datasets, logging systems, administrative interfaces, or security configurations. Excessive permissions can allow insiders to copy training data, access sensitive prompts, alter retrieval indexes, or export model weights.

ISSN: 3071-5717
PP: 1-32.

Third-party applications and plugins can also become threat actors when compromised or poorly controlled. Generative AI systems increasingly interact with external tools, search functions, enterprise applications, and automated workflows. A compromised tool can provide malicious instructions to the model or exploit the model's access to other resources. Greshake et al. (2023) showed that indirect prompt injection can exploit untrusted external content to influence the behavior of language-model applications.

AI agents create an additional category of risk because they may operate with persistent memory and access to external tools. Wang et al. (2025) identified privacy concerns associated with agent memory, showing that sensitive information can persist beyond a single interaction. This increases the need for access restrictions on what agents can store, retrieve, and share.

4.3. Attack Surfaces

The attack surface of a generative AI system spans the complete data and model lifecycle. During data ingestion, attackers may introduce poisoned, manipulated, or unauthorized documents. In RAG systems, malicious documents can be designed to achieve high retrieval relevance and influence generated responses. Zou et al. (2025) demonstrated that poisoning attacks can corrupt retrieved knowledge and alter the output of retrieval-augmented generation systems.

During model training and fine-tuning, sensitive information may be unintentionally memorized. Attackers may later exploit this through training-data extraction, membership inference, or model inversion. Membership inference attempts to determine whether a specific record was used during training, while model inversion aims to reconstruct sensitive attributes from model behavior. Shokri et al. (2017) established the practical feasibility of membership inference against machine-learning models, and Fredrikson et al. (2015) showed how model outputs can support inversion attacks.

Prompt interfaces constitute another major attack surface. Attackers may use direct or indirect prompt injection to bypass restrictions, reveal hidden context, or manipulate connected tools. The risk increases when prompts are combined with RAG, autonomous agents, plugins, or enterprise APIs.

Model APIs and deployment endpoints can be targeted through credential theft, excessive querying, model extraction, denial of service, or unauthorized fine-tuning. Tramèr et al. (2016) demonstrated that prediction APIs can be used to reproduce machine-learning models through repeated queries. Carlini et al. (2024) later showed that information about production language models can also be recovered through carefully constructed interactions.

Output channels represent the final major attack surface. Even when access to the system is legitimate, generated responses may disclose restricted information. Without output inspection, a model may expose personal data, credentials, proprietary text, or retrieved documents. Logs and

ISSN: 3071-5717
PP: 1-32.

monitoring systems can also become secondary exposure points if prompts and responses are retained without suitable controls.

A data-centric threat model must therefore treat the AI system as a chain of interconnected exposure points rather than a single protected application. Security decisions should remain linked to the sensitivity of the data as it moves through each component.

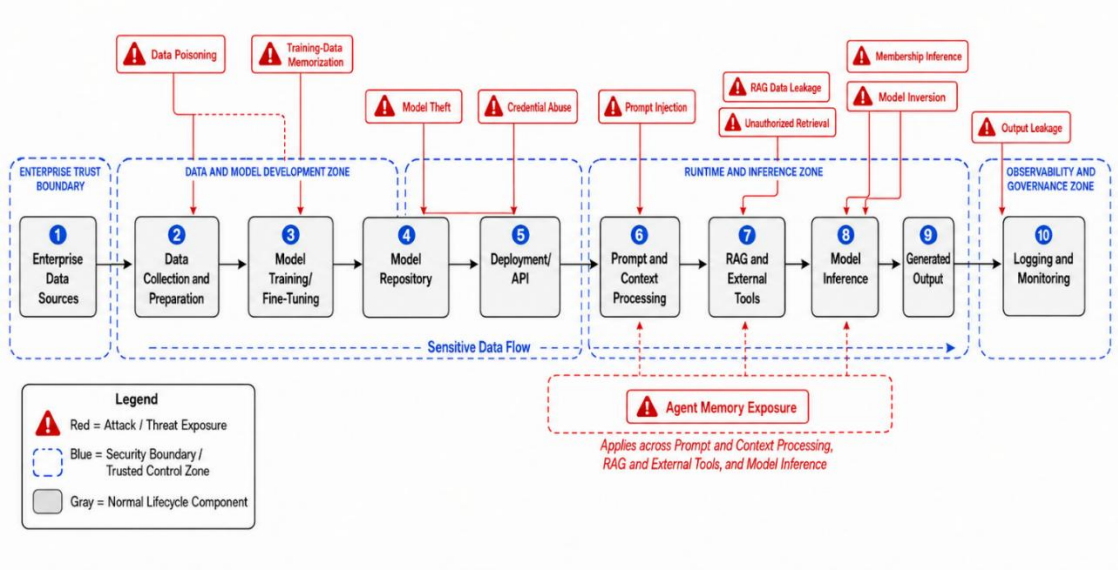

Fig 1 - Data Exposure and Attack Surfaces across the Generative AI Lifecycle

5. DATA-CENTRIC SECURITY FRAMEWORK FOR GENERATIVE AI

The proposed data-centric security framework places sensitive information at the center of protection decisions throughout the generative AI lifecycle. The central principle is that protection should follow the data rather than depend solely on the security of the infrastructure where the data is stored. A document classified as restricted, for example, should remain subject to the same protection requirements when it is converted into an embedding, retrieved into a model context, processed during inference, or referenced in a generated response.

The framework combines data governance, privacy protection, identity control, secure retrieval, model protection, output inspection, and continuous monitoring. These controls are arranged as connected layers rather than isolated mechanisms.

5.1. Data Discovery and Classification

Effective protection begins with understanding what information enters the generative AI environment. Organizations should identify datasets, documents, prompts, logs, embeddings, and model artifacts that contain sensitive information. Data classification should define categories such as public, internal, confidential, and restricted.

ISSN: 3071-5717
PP: 1-32.

Classification should influence whether information can be used for training, fine-tuning, retrieval, external processing, logging, or model output. Data owners should also define retention periods, permitted users, and acceptable purposes for each category.

5.2. Data Minimization and Sanitization

Generative AI systems should receive only the information necessary for their intended task. Unnecessary personal identifiers, credentials, confidential fields, or duplicate records should be removed before training or retrieval.

Data minimization can be supported through anonymization, pseudonymization, masking, tokenization, redaction, and deduplication. Kandpal et al. (2022) found that training-data deduplication can reduce privacy risks associated with language-model memorization. This indicates that data preparation itself is an important security control rather than merely a preprocessing activity.

5.3. Cryptographic Protection

Sensitive information should be encrypted during storage and transmission. Training datasets, model checkpoints, vector databases, logs, API secrets, and backups should be protected by strong cryptographic controls.

Encryption keys should be managed separately from the systems that store the protected data. Access to keys should be limited according to least-privilege principles. Confidential computing and trusted execution environments may provide additional protection where sensitive information must be processed in memory.

5.4. Privacy-Preserving Model Training

Privacy-preserving methods can reduce the risk that models retain identifiable training examples. Differential privacy is particularly relevant because it introduces controlled noise during learning to limit the influence of individual records. Abadi et al. (2016) established differential privacy as a practical approach for deep learning, while Li et al. (2022) showed that large language models can also be trained under differential privacy constraints.

Other approaches include federated learning, secure multiparty computation, confidential training environments, and controlled fine-tuning. These methods involve trade-offs between privacy, computational cost, and model performance, which should be assessed according to the sensitivity of the training data.

5.5. Identity-Centric Model and Data Access

Access to generative AI services should be based on verified identity, role, context, and resource sensitivity. Authentication alone is insufficient because different users may require different access to the same AI system.

ISSN: 3071-5717
PP: 1-32.

Role-based access control can assign permissions according to organizational roles, while attribute-based access control can consider factors such as department, clearance level, location, device status, or data classification. Privileged users should be subject to additional controls through privileged access management and stronger authentication.

Model access and data access should also be separated. A user who can access a generative AI application should not automatically gain access to every repository available to that application.

5.6. Prompt and Context Security

Prompts should be treated as untrusted input. Input inspection can identify suspicious instructions, attempts to override system rules, requests for hidden information, or content designed to manipulate tool behavior.

Prompt security should also maintain clear separation between system instructions, user inputs, retrieved documents, and tool outputs. Retrieved content should not be treated as trusted instructions simply because it enters the model context. Greshake et al. (2023) demonstrated why this distinction is necessary in systems exposed to indirect prompt injection.

Context isolation is particularly important in multi-user environments. Information from one user session should not appear in another user's context unless explicitly authorized.

5.7. Secure Retrieval-Augmented Generation

RAG systems should enforce authorization before information is retrieved, not only after a response has been generated. Each document should retain access-control metadata, and retrieval should be filtered according to the identity and permissions of the requesting user.

Additional controls should include tenant isolation, source validation, provenance verification, vector database encryption, document classification, retrieval logging, and poisoning detection. Zeng et al. (2024) identified privacy risks associated with RAG systems, while Zou et al. (2025) demonstrated that malicious documents can manipulate retrieval behavior. These findings support the need to secure both access and integrity within the retrieval layer.

5.8. Output Protection and Data Loss Prevention

Generated responses should be inspected before reaching the user or external system. Output controls can identify personal information, authentication credentials, confidential text, internal identifiers, or restricted documents.

A practical control path is:

Model Output → Sensitive Data Detection → Policy Evaluation → Allow / Redact / Block / Escalate

ISSN: 3071-5717
PP: 1-32.

Data loss prevention rules should be linked to the same classification policies applied to source information. Where the system is uncertain about the sensitivity of an output, high-risk responses can be routed for human review.

5.9. Model and API Protection

Model endpoints should be protected through strong authentication, API gateways, rate limiting, short-lived credentials, secrets management, and behavioural monitoring. Repeated or unusual queries may indicate extraction attempts, automated abuse, or efforts to recover memorized data.

Model repositories should also require strict access controls because unauthorized access to model weights can expose intellectual property and make offline attacks easier. Carlini et al. (2024) showed that production language models can reveal sensitive model information through interaction, reinforcing the need for query monitoring and endpoint protection.

5.10. Continuous Monitoring and Auditability

Security controls should be supported by continuous monitoring of prompts, retrieval events, data-access requests, API calls, generated responses, authentication activity, model changes, and administrative actions. Logging should support incident investigation without creating another sensitive-data repository. Organizations should therefore avoid storing unnecessary prompt or response content and should apply retention limits, encryption, and access restrictions to security logs.

Monitoring should also identify unusual patterns such as repeated attempts to retrieve restricted data, abnormal prompt sequences, unusually high query volumes, repeated requests for memorized content, or access from unfamiliar identities. Audit records should provide enough information to determine who accessed a model, what data sources were used, and what controls were applied. Taken together, these controls establish a security model in which sensitive information remains governed across its full movement through the generative AI system. The framework does not rely on a single preventive mechanism. Instead, it combines data classification, privacy protection, identity control, retrieval authorization, output filtering, and monitoring to reduce the likelihood that one failed control results in sensitive-data exposure.

ISSN: 3071-5717
PP: 1-32.

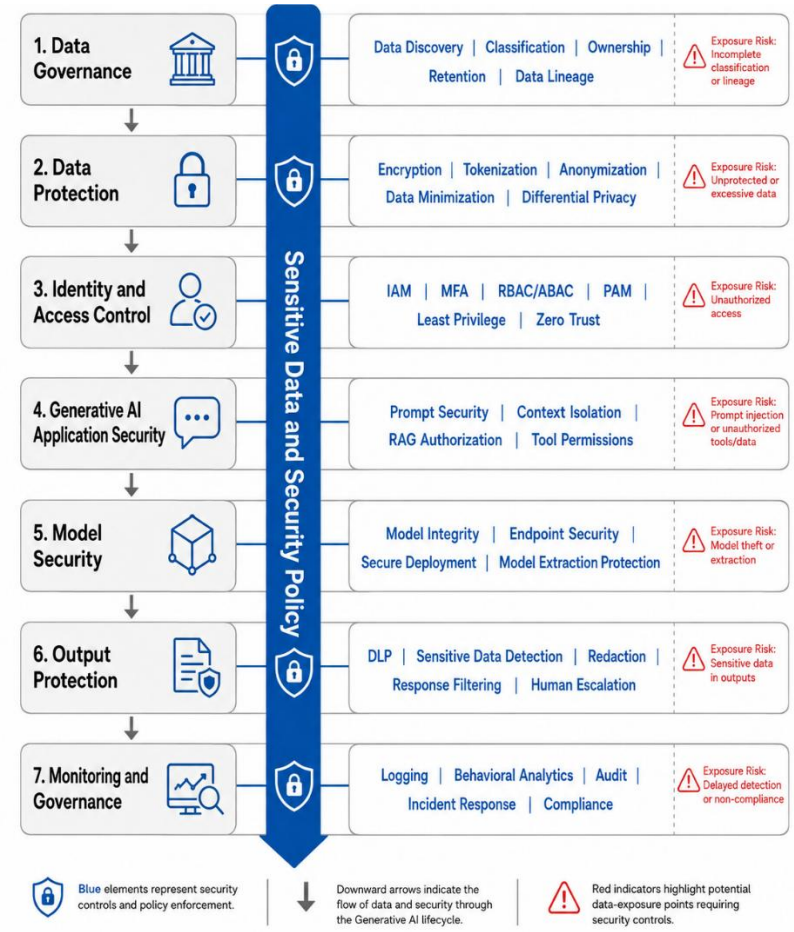

Fig 2 - Proposed Layered Data-Centric Security Framework for Protecting Sensitive Information across Generative AI Systems

6. SECURITY CONTROLS AGAINST MAJOR ATTACK CLASSES

Security controls for generative AI should be designed around the specific ways in which sensitive information can be exposed, inferred, extracted, or accessed without authorization. No single control can address all attack classes because the underlying mechanisms differ considerably. Training-data leakage arises from memorization, prompt-based attacks exploit instruction-following behavior, retrieval attacks target external knowledge sources, and model extraction depends on repeated interaction with accessible endpoints. Effective protection therefore requires controls that are matched to each threat while remaining connected through a common data-classification and access policy.

6.1. Preventing Training-Data Leakage

Training-data leakage occurs when a model reveals information that originated from its training or fine-tuning datasets. The risk is greatest when sensitive records are rare, repeated, highly distinctive, or insufficiently filtered before training. Carlini et al. (2021) showed that large language models can reproduce memorized training sequences, while Carlini et al. (2023) demonstrated that

ISSN: 3071-5717
PP: 1-32.

memorization is influenced by model scale, data duplication, and the characteristics of individual examples. These findings indicate that protecting the original training dataset is not enough once sensitive information has been incorporated into a model.

The first control should therefore be applied before training begins. Organizations should minimize the use of sensitive information, remove unnecessary identifiers, exclude credentials and confidential records, and apply clear rules governing which datasets may be used for model development. Data deduplication is also important because repeated records can increase the probability that specific examples are memorized. Kandpal et al. (2022) found that deduplicating language-model training data can reduce privacy risk, supporting its use as a practical preprocessing measure.

Differential privacy provides an additional layer of protection by limiting the influence of individual training records. Abadi et al. (2016) demonstrated how differential privacy can be incorporated into deep learning, while Li et al. (2022) showed that large language models can retain useful capabilities under differentially private learning conditions. The value of differential privacy is that it provides a formal mechanism for reducing the likelihood that the presence of a particular individual record materially changes model behavior.

Training controls should also include memorization testing before deployment. Models can be evaluated using canary records, exposure measurements, targeted extraction tests, and privacy audits. When a model shows a tendency to reproduce sensitive or distinctive examples, deployment should be delayed until the issue is addressed through additional filtering, retraining, privacy controls, or changes to the training dataset.

6.2. Defending Against Membership and Attribute Inference

Membership inference attacks attempt to determine whether a particular record was included in a model's training data. Shokri et al. (2017) showed that model outputs can reveal information about training-set membership, establishing an important privacy risk that remains relevant to modern generative systems. Attribute inference presents a related problem in which an adversary uses available model information to infer sensitive characteristics that are not directly disclosed.

A central defense is to reduce unnecessary information in model responses. Systems should avoid exposing internal confidence values, detailed probability distributions, or other signals unless they are required for legitimate use. Excessive output detail can give attackers additional information that improves inference accuracy.

Differential privacy can also reduce membership risk by limiting the influence of individual records during training. However, privacy protection should not rely on training controls alone. Query monitoring and rate limiting are necessary because many inference attacks require repeated interaction

ISSN: 3071-5717
PP: 1-32.

with a target model. A sequence of highly similar prompts targeting the same person, document, or phrase may indicate an attempt to determine membership or infer private attributes.

Access controls should also consider the sensitivity of the model itself. Models trained or fine-tuned on confidential organizational data should not be exposed through broadly accessible interfaces. Restricting access according to role, purpose, and operational need reduces the number of users capable of conducting systematic inference attacks.

6.3. Defending Against Model Inversion and Extraction

Model inversion attacks attempt to reconstruct information about training records or sensitive attributes from model behavior. Fredrikson et al. (2015) demonstrated that confidence information can support reconstruction attacks, showing that model outputs may reveal more about underlying data than intended.

The first line of defense is output minimization. Models should return only the information required to satisfy legitimate requests. Where applications expose structured prediction scores, embeddings, or internal representations, organizations should assess whether that information creates unnecessary reconstruction risk.

Model extraction presents a separate but related threat. Tramèr et al. (2016) showed that prediction APIs can be used to reproduce machine-learning models through repeated querying. More recent work by Carlini et al. (2024) demonstrated that production language models can also reveal significant information about their internal structure through carefully designed interactions.

Controls against extraction should include query-rate restrictions, usage quotas, anomaly detection, strong authentication, and monitoring for repeated probing patterns. Requests that systematically vary inputs while examining small changes in outputs may indicate an attempt to map model behavior. High-volume automated querying should therefore be examined differently from normal conversational use.

Model repositories require equally strict protection. Encryption, access logging, privileged access management, and separation of administrative roles can reduce the risk of direct model theft. Model weights should be treated as high-value intellectual property rather than ordinary application files.

6.4. Preventing Prompt-Based Leakage

Prompt injection is one of the most direct ways to manipulate generative AI applications. Attackers may attempt to override application instructions, reveal system prompts, access hidden context, extract retrieved documents, or influence connected tools. Indirect prompt injection is

ISSN: 3071-5717
PP: 1-32.

particularly difficult because malicious instructions can be embedded within websites, documents, emails, or other content processed by the model.

Greshake et al. (2023) demonstrated that indirect prompt injection can compromise real-world language-model applications by causing models to treat untrusted external content as instructions. This problem becomes more serious when a model can access enterprise data or take actions through external tools.

Defenses should begin with clear separation of system instructions, user prompts, retrieved data, and tool outputs. Content retrieved from external sources should be treated as data, not as trusted instructions. Prompt filtering can identify known attack patterns, but filtering alone is insufficient because attackers can alter wording or embed instructions indirectly.

Tool access should therefore follow least-privilege principles. A model that only needs document retrieval should not receive unrestricted access to email, databases, payment systems, or administrative functions. Sensitive actions should require explicit authorization, and high-risk operations may require human approval.

Context isolation is also important. Information from one user, department, or session should not be available to another unless access has been deliberately granted. Output inspection should provide a final control to detect restricted information before it reaches the user.

6.5. Preventing Unauthorized Model Access

Unauthorized access may result from stolen passwords, exposed API keys, weak authentication, excessive privileges, insecure service accounts, or poorly controlled model repositories. Once an attacker gains access, the model can become an entry point to sensitive information and connected enterprise resources.

Strong authentication should be required for administrative interfaces, model repositories, and high-risk AI services. Multi-factor authentication is particularly important for privileged accounts. API keys should be stored in dedicated secrets-management systems rather than embedded in application code or shared through configuration files.

Authorization should follow least privilege. Users, applications, and agents should receive only the permissions required for their assigned tasks. Role-based and attribute-based access control can support this by considering the user's role, department, clearance, device, location, and sensitivity of the requested data.

ISSN: 3071-5717
PP: 1-32.

Short-lived credentials can reduce the impact of stolen tokens, while regular credential rotation can limit long-term misuse. API gateways can enforce authentication, rate limits, request validation, and usage policies before traffic reaches the model.

Behavioral monitoring should complement these measures. A valid credential does not guarantee legitimate use. Sudden changes in query volume, access location, requested data, or tool usage may indicate account compromise or insider misuse.

7. GOVERNANCE, RISK MANAGEMENT, AND COMPLIANCE

Technical controls alone cannot provide lasting protection if responsibility for data, models, and access decisions is unclear. Generative AI security therefore requires governance structures that define ownership, acceptable use, risk thresholds, monitoring responsibilities, and accountability. Governance should connect security decisions with the sensitivity of the information being processed rather than treating all generative AI applications as equivalent.

7.1. Generative AI Data Governance

Data governance should begin with clear ownership. Every dataset used for training, fine-tuning, retrieval, or evaluation should have an identified owner responsible for determining its sensitivity, permitted use, retention period, and access conditions. Model owners should be responsible for deployment, configuration, testing, and model-specific risks, while application owners should oversee how models are connected to users, data sources, and external tools.

Security teams should define technical requirements for authentication, encryption, monitoring, incident response, and vulnerability management. Privacy teams should evaluate whether personal information is collected, retained, or processed beyond its intended purpose. Business owners should determine whether the use of a particular data source is necessary and proportionate to the operational objective.

This division of responsibility is important because many security failures occur when sensitive data passes between teams without clear accountability. A dataset may be approved for internal analysis but later incorporated into a RAG system or fine-tuning process without renewed review. Governance should therefore require reassessment when the purpose, model, user population, or data flow changes.

Data lineage is also important. Organizations should be able to determine where information came from, which models or indexes received it, who accessed it, and whether it appears in derived datasets, embeddings, or logs. Without this information, incident investigation and deletion requests become difficult.

ISSN: 3071-5717
PP: 1-32.

7.2. Policy-Based AI Data Usage

Organizations should establish explicit rules describing which information may be processed by generative AI systems. These policies should distinguish between public, internal, confidential, and restricted data rather than applying a single rule to all information.

For example, public information may be suitable for broad model use, while confidential information may require approved enterprise-hosted models, encryption, restricted retrieval, and logging controls. Restricted information may be prohibited from external model services or permitted only within tightly controlled environments.

Policies should also address the use of data for training and fine-tuning. Data that is appropriate for one-time inference may not be appropriate for permanent incorporation into a model. Similarly, information that can be retrieved under user-specific permissions should not automatically be added to a shared training dataset.

Prompt usage policies should explain whether users may submit customer records, source code, legal documents, financial information, credentials, or other sensitive material. Clear rules reduce uncertainty and help prevent accidental disclosure.

Policies should extend to AI agents. Agents with access to external tools should have defined limits on what information they can retrieve, store, transmit, or act upon. Persistent memory should be governed by retention and deletion requirements rather than allowed to expand without control.

7.3. AI Data Risk Assessment

Risk assessment should evaluate both the sensitivity of the information and the ways in which the generative AI system may expose it. A useful assessment can consider four dimensions:

Data Sensitivity × Exposure Probability × Attack Feasibility × Business Impact

Data sensitivity reflects the consequences of unauthorized disclosure. Exposure probability considers how easily information can reach users, models, tools, logs, or external systems. Attack feasibility evaluates whether a realistic adversary could exploit the system through prompts, inference, retrieval manipulation, stolen credentials, or repeated queries. Business impact considers financial, legal, operational, reputational, and intellectual-property consequences.

A model trained on public information and used for general writing assistance carries a different risk profile from a model connected to confidential legal, financial, or customer records. Security requirements should reflect these differences.

Risk assessments should also consider the attack surface created by connected tools and retrieval systems. A model with no external access may have limited consequences if manipulated. The

ISSN: 3071-5717
PP: 1-32.

same model connected to email, document repositories, databases, and automated actions presents a substantially greater risk.

Assessments should be repeated when models are upgraded, new data sources are connected, permissions change, agents receive new tools, or applications are made available to larger user groups.

7.4. Continuous AI Security Testing

Generative AI systems require repeated security testing because model behavior can change across versions, prompts, data sources, and deployment configurations. Traditional vulnerability scanning remains necessary, but it should be supplemented with tests that reflect model-specific threats.

Security teams should conduct prompt-injection testing, training-data extraction attempts, membership inference assessments, retrieval authorization tests, model-extraction simulations, and checks for sensitive output disclosure. Greshake et al. (2023) provides evidence of the importance of testing indirect prompt injection, while Nasr et al. (2025) shows why extraction testing remains relevant even for aligned production language models.

RAG systems should be tested to ensure that users cannot retrieve documents outside their permission scope. Test cases should include cross-department access attempts, malicious document insertion, poisoned retrieval sources, and requests designed to reconstruct restricted content. Zeng et al. (2024) and Zou et al. (2025) demonstrate why privacy and retrieval integrity should both be included in security evaluation.

Red-team exercises can also examine how several weaknesses combine. An attacker may first use prompt injection to influence model behavior, then exploit tool access to retrieve information, and finally use output channels to exfiltrate it. Testing isolated controls may miss these attack chains.

7.5 Regulatory and Standards Alignment

Governance for generative AI should align with the privacy, information-security, sector-specific, and data-protection requirements that already apply to organizational information. Introducing a generative model does not remove existing obligations concerning confidentiality, lawful processing, access control, retention, incident reporting, or accountability.

Organizations should therefore map generative AI data flows to their existing compliance requirements. This includes identifying where personal or confidential information is collected, where it is stored, whether it is transferred to third parties, how long it is retained, and who can access it.

Model providers and external AI services should also be subject to supplier-risk assessment. Organizations need to understand whether submitted prompts are retained, whether data may be used

ISSN: 3071-5717
PP: 1-32.

for service improvement, where processing occurs, how access is controlled, and what deletion mechanisms are available. These questions should form part of procurement and security review rather than being addressed after deployment.

Compliance evidence should be supported by logs, access records, model inventories, data lineage, risk assessments, approval records, and incident documentation. These records make it possible to demonstrate that controls are not merely described in policy but are applied in practice.

Governance should also include a process for responding to security incidents involving generative AI. An incident may require revoking credentials, disabling model access, removing a compromised retrieval source, isolating an agent, reviewing logs, deleting exposed data, or notifying affected stakeholders. Clear responsibilities should be established before an incident occurs.

Taken together, governance, risk management, and compliance provide the organizational structure required to support the technical controls described in Section 6. Data-centric security becomes effective when organizations know what information they hold, understand how it is used by generative AI systems, restrict access according to sensitivity and purpose, test the controls regularly, and maintain evidence that these protections are operating as intended.

8. DISCUSSION

The findings of this study show that data protection in generative AI cannot be treated as a secondary extension of conventional cybersecurity. Generative systems interact with information in ways that create new forms of exposure across training, retrieval, inference, output generation, and persistent memory. Sensitive data may be reproduced directly, inferred indirectly, retrieved beyond the user's authorization scope, or exposed through model interfaces that were not originally designed as data-access channels. These risks explain why security controls must remain attached to the data throughout its lifecycle rather than being limited to network, application, or storage boundaries.

8.1. Key Findings

One of the central findings is that sensitive information can remain exposed even after the original dataset has been secured. Research on memorization and training-data extraction demonstrates that trained models can retain information about individual records and, under certain conditions, reproduce parts of their training data. Carlini et al. (2021) showed that large language models can reveal memorized training sequences, while Carlini et al. (2023) further demonstrated that memorization is affected by model scale, duplication, and properties of the underlying data. These findings confirm that model security and data security cannot be separated.

A second finding concerns the growing importance of retrieval-augmented generation. RAG systems reduce dependence on static training data by retrieving external information at inference time, but they also create a direct path between generative models and enterprise knowledge repositories.

ISSN: 3071-5717
PP: 1-32.

Zeng et al. (2024) showed that retrieval mechanisms introduce privacy risks when models interact with external knowledge sources. These risks are amplified when document-level authorization is not consistently enforced or when poisoned documents influence what the model retrieves. Zou et al. (2025) demonstrated that malicious content can manipulate RAG behavior, highlighting the need to protect both confidentiality and retrieval integrity.

A third finding is that the user interface itself can become an attack surface. Prompt injection, indirect prompt injection, repeated extraction attempts, and model probing all exploit legitimate model functionality rather than conventional software vulnerabilities. Greshake et al. (2023) demonstrated that indirect prompt injection can compromise systems that process external content, while Carlini et al. (2024) showed that carefully designed interactions can reveal substantial information about production language models. Security must therefore account for how models respond to valid-looking but malicious inputs.

8.2 Why Traditional Security Is Insufficient

Traditional security frameworks usually emphasize authentication, authorization, perimeter defenses, network segmentation, endpoint protection, and secure storage. These controls remain essential, but they do not fully address the behavior of generative models. A user may be correctly authenticated and still cause sensitive information to be exposed through a prompt, retrieval request, or repeated inference attack.

The distinction between system access and data access is therefore critical. Granting permission to use an AI application should not imply permission to access every document, embedding, memory record, or connected data source available to the model. Generative AI security requires security decisions to consider the identity of the user, the classification of the requested information, the purpose of the request, and the sensitivity of the output.

This is where a data-centric approach offers greater protection. Instead of assuming that security is achieved once a user is inside an approved application, the proposed framework requires continued enforcement of data permissions throughout training, retrieval, inference, and response generation. This approach can reduce the risk that information loses its protection simply because it has moved into a different technical representation.

8.3. Security and Utility Trade-Off

Data-centric protection introduces trade-offs that organizations must manage carefully. Strong privacy controls can sometimes reduce model utility, increase computational requirements, or create additional latency. Differential privacy, for example, can reduce information leakage by limiting the influence of individual records, but stronger privacy guarantees may affect model accuracy or require additional training resources. Li et al. (2022) showed that large language models can be trained under

ISSN: 3071-5717
PP: 1-32.

differential privacy, but practical deployment still requires careful balancing of privacy and performance.

Similar trade-offs appear in output filtering, access control, retrieval restrictions, and human review. Excessive filtering can reduce usefulness by blocking legitimate requests, while weak filtering can expose sensitive content. Strict retrieval authorization may limit the information available to the model, but insufficient authorization creates confidentiality risks.

The objective should therefore not be maximum restriction. Effective security requires proportionate controls based on data sensitivity, operational purpose, attack likelihood, and potential impact. High-risk data should receive stronger controls, while low-risk applications may operate with lighter safeguards.

8.4. Practical Implications

The proposed framework has several practical implications for organizations deploying generative AI. First, security teams should treat generative AI as part of the organization's broader data-governance environment rather than as an isolated technology. Existing classification, identity, encryption, retention, and monitoring controls should be extended into AI pipelines.

Second, organizations should require authorization at the point of retrieval. A model should not retrieve documents simply because they are indexed in a vector database. Retrieval systems should enforce the same user and document permissions that apply in the original source environment.

Third, model and API monitoring should examine behavioral patterns rather than relying only on authentication events. Repeated probing, unusual query volumes, persistent attempts to reconstruct sensitive data, or sudden changes in tool usage may indicate extraction or abuse even when valid credentials are used.

Finally, organizations should incorporate generative AI into existing incident-response procedures. Security teams should be able to revoke model access, disable compromised data sources, isolate agents, rotate credentials, review retrieval histories, and investigate output logs when suspicious activity is detected.

8.5. Theoretical Contributions

This study contributes to the literature by bringing together concepts that are often examined separately, including data security, privacy-preserving machine learning, Zero Trust, model protection, prompt security, RAG authorization, and AI governance. The proposed framework treats these controls as parts of a single data-centric security architecture.

ISSN: 3071-5717
PP: 1-32.

The central theoretical contribution is the view that sensitive information should remain governed by consistent security policies regardless of whether it exists as raw data, model training material, an embedding, retrieved context, model memory, or generated output. This principle extends conventional data-centric security into generative AI environments, where information frequently changes form while retaining the same underlying sensitivity.

9. CHALLENGES AND FUTURE RESEARCH

Despite progress in generative AI security, several unresolved challenges remain. Many existing controls address individual attacks, but fewer studies examine how these controls operate together in complex enterprise environments. Future research should therefore focus on integrated protection across models, retrieval systems, agents, external tools, and persistent memory.

One major challenge is reliable measurement of model memorization. Existing extraction methods can demonstrate that memorization occurs, but it remains difficult to determine how much sensitive information a deployed model retains or whether all risky examples have been identified. Future work should develop standardized privacy-testing procedures that can be applied before deployment and repeated after fine-tuning or model updates.

Embedding privacy is another important research area. RAG systems increasingly depend on vector representations of confidential documents, yet the privacy characteristics of embeddings remain less understood than traditional database protection. Research is needed to determine how embeddings may reveal sensitive information, how they can be protected without reducing retrieval quality, and how authorization metadata should remain associated with embedded content.

Agent memory also deserves greater attention. As generative systems become more autonomous, they increasingly store user preferences, previous interactions, intermediate decisions, and retrieved information. Wang et al. (2025) showed that agent memory can create privacy risks, suggesting that long-term memory should be treated as a protected data store rather than a simple usability feature. Future research should examine secure memory isolation, expiration, deletion, encryption, and cross-agent information sharing.

Multi-agent systems create additional concerns. Several agents may exchange information, delegate tasks, call tools, and maintain separate or shared memory. Sensitive data can therefore move across multiple autonomous components without direct user supervision. Future security models should address inter-agent authorization, provenance, delegation controls, and mechanisms for preventing one compromised agent from accessing the resources of another.

Another area for research is machine unlearning. Organizations may be required to remove personal or proprietary information after a model has already been trained. Deleting the original dataset does not guarantee that the information has been removed from model behavior. Future work

ISSN: 3071-5717
PP: 1-32.

should develop more reliable methods for removing specific information from trained models and verifying that the removal has been successful.

Confidential inference and secure model execution also remain important. Techniques such as trusted execution environments, secure multiparty computation, and homomorphic encryption could reduce exposure during model processing, but their computational cost can limit practical use. Research is needed to improve the efficiency of these methods for large-scale generative models.

Future work should also examine continuous authorization for autonomous AI agents. Traditional access controls often make decisions when a user initially connects to a system. Autonomous agents, however, may operate for extended periods, access multiple tools, and act under changing conditions. Authorization should therefore be reassessed as tasks, data sensitivity, and operational context change.

Finally, stronger empirical evaluation is required. Many proposed defenses are evaluated against a limited set of models or attack scenarios. Future studies should compare security controls across different model sizes, architectures, RAG configurations, deployment environments, and attacker capabilities. Such evaluations would help identify controls that remain effective under realistic enterprise conditions.

10. CONCLUSION

Generative AI has changed the way sensitive information is stored, processed, retrieved, and disclosed. Data may pass through training datasets, model parameters, prompts, embeddings, retrieval systems, agent memory, outputs, and logs, creating exposure points that conventional security controls may not fully address. Training-data extraction, membership inference, model inversion, prompt injection, RAG leakage, model extraction, and unauthorized access demonstrate that the security problem extends beyond protecting infrastructure alone.

This study proposed a data-centric security framework that places sensitive information at the center of protection decisions throughout the generative AI lifecycle. The framework combines data discovery and classification, minimization, cryptographic protection, privacy-preserving training, identity and access control, prompt security, secure RAG, output protection, model and API security, continuous monitoring, and governance. The main principle is that security requirements should remain attached to information regardless of where that information is stored or how it is represented. A confidential document should not lose its protection when converted into an embedding, retrieved into a prompt context, processed by a model, or referenced in a generated response.

Protecting generative AI therefore requires more than stronger authentication or safer models. It requires coordinated control over data, models, users, retrieval systems, agents, and outputs. A data-centric approach provides a practical foundation for reducing leakage, limiting inference attacks,

ISSN: 3071-5717
PP: 1-32.

preventing unauthorized model access, and supporting accountable use of generative AI in environments where sensitive information is central to everyday operations.

REFERENCES

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318. doi:10.1145/2976749.2978318.

[2] Aerni, M., Rando, J., Debenedetti, E., Carlini, N., Ippolito, D., & Tramèr, F. (2025). Measuring non-adversarial reproduction of training data in large language models. *International Conference on Learning Representations (ICLR 2025)*.

[3] An, B., Zhang, S., & Dredze, M. (2025). RAG LLMs are not safer: A safety analysis of retrieval-augmented generation for large language models. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5444–5474. doi: 10.18653/v1/2025.naacl-long.281.

[4] Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., & Song, D. (2019). The Secret Sharer: Evaluating and testing unintended memorization in neural networks. *28th USENIX Security Symposium (USENIX Security 19)*, 267–284.

[5] Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., & Raffel, C. (2021). Extracting training data from large language models. *30th USENIX Security Symposium (USENIX Security 21)*, 2633–2650.

[6] Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., & Zhang, C. (2023). Quantifying memorization across neural language models. *International Conference on Learning Representations (ICLR 2023)*.

[7] Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., & Wallace, E. (2023). Extracting training data from diffusion models. *32nd USENIX Security Symposium (USENIX Security 23)*, 5253–5270.

[8] Carlini, N., Paleka, D., Dvijotham, K. D., Steinke, T., Hayase, J., Cooper, A. F., Lee, K., Jagielski, M., Nasr, M., Conmy, A., Wallace, E., Rolnick, D., & Tramèr, F. (2024). Stealing part of a production language model. *Proceedings of the 41st International Conference on Machine Learning*, 235, 5680–5705.

[9] Kunaparaju, C. (2025). AI-Driven Cyber Defense Systems: Strengthening National Security through Intelligent Threat Prediction and Response. Algora, 2(1), 1-30.

[10] Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G. J., Tramèr, F., Hassani, H., & Wong, E. (2024). JailbreakBench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems, 37*. doi:10.52202/079017-1745.

[11] Potla, R. B. (2024). Optimizing extended warehouse management for make-to-order plants: Slotting, wave picking, and yard orchestration at scale. Journal of Computer Science and Technology Studies, 6(3), 181-192.

[12] Cheng, Y., Zhang, L., Wang, J., Yuan, M., & Yao, Y. (2025). RemoteRAG: A privacy-preserving LLM cloud RAG service. *Findings of the Association for Computational Linguistics: ACL 2025*, 3820–3837. doi: 10.18653/v1/2025.findings-acl.197.

[13] Flemings, J., Jiang, B., Zhang, W., Takhirov, Z., & Annavaram, M. (2025). Estimating privacy leakage of augmented contextual knowledge in language models. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 25092–25108. doi: 10.18653/v1/2025.acl-long.1220.

[14] Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333. doi:10.1145/2810103.2813677.

[15] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. doi:10.1145/3605764.3623985.

[16] Huang, J., Shao, H., & Chang, K. C.-C. (2022). Are large-pre-trained language models leaking your personal information? *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2038–2047. doi: 10.18653/v1/2022.findings-emnlp.148.

[17] Huang, Y., Gupta, S., Zhong, Z., Li, K., & Chen, D. (2023). Privacy implications of retrieval-based language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 14887–14902.

ISSN: 3071-5717
PP: 1-32.

[18] Potla, R. B. (2024). A SOX/ITAR-aligned global ERP template for multi-plant manufacturers: Governance patterns and controls. J Artif Intell Mach Learn & Data Sci, 2(2), 3222-3232.

[19] Begimher, D., Leo, C., Huang, J., Gaw, P., & Zheng, B. (2026). SIR-Bench: Evaluating Investigation Depth in Security Incident Response Agents. arXiv preprint arXiv:2604.12040.

[20] Kandpal, N., Wallace, E., & Raffel, C. (2022). Deduplicating training data mitigates privacy risks in language models. *Proceedings of the 39th International Conference on Machine Learning, 162*, 10697–10707.

[21] Li, X., Tramèr, F., Liang, P., & Hashimoto, T. (2022). Large language models can be strong differentially private learners. *International Conference on Learning Representations (ICLR 2022)*.

[22] Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., & Zanella-Béguelin, S. (2023). Analyzing leakage of personally identifiable information in language models. *2023 IEEE Symposium on Security and Privacy*, 346–363.

[23] Nasr, M., Shokri, R., & Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Symposium on Security and Privacy*, 739–753.

[24] Nasr, M., Rando, J., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Tramèr, F., & Lee, K. (2025). Scalable extraction of training data from aligned, production language models. *International Conference on Learning Representations (ICLR 2025)*.

[25] Perez, F., & Ribeiro, I. (2022). Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*.

[26] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy*, 3–18. doi:10.1109/SP.2017.41.

[27] Kunaparaju, C. (2024). The Role of Artificial Intelligence in Safeguarding Critical National Infrastructure against Cyberattacks. Journal of Electrical Systems, 20, 5403-5419.

[28] Somepalli, G., Singla, V., Goldblum, M., Geiping, J., & Goldstein, T. (2023). Diffusion art or digital forgery? Investigating data replication in diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[29] Potla, R. (2023). Designing a BTP-centric integration mesh for shop-floor IoT, MES and ERP in discrete manufacturing. Journal of Artificial Intelligence, Machine Learning and Data Science, 1(2), 1-8.

[30] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. *25th USENIX Security Symposium (USENIX Security 16)*, 601–618.

[31] Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., Truong, S., Arora, S., Mazeika, M., Hendrycks, D., Lin, Z., Cheng, Y., Koyejo, S., Song, D., & Li, B. (2023). DecodingTrust: A comprehensive assessment of trustworthiness in GPT models. *Advances in Neural Information Processing Systems, 36*.

[32] Wang, B., He, W., Zeng, S., Xiang, Z., Xing, Y., Tang, J., & He, P. (2025). Unveiling privacy risks in LLM agent memory. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 25241–25260. doi: 10.18653/v1/2025.acl-long.1227.

[33] Leo, C., Dykyi, A., Cortegaca, D., Begimher, D., & Jha, P. (2026). ThreatForest: Multi-Agent Attack Tree Generation with Pluggable TTP Framework Mapping. arXiv preprint arXiv:2607.27528.

[34] Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? *Advances in Neural Information Processing Systems, 36*.

[35] Yi, J., et al. (2025). Benchmarking and defending against indirect prompt injection attacks on large language models. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. doi:10.1145/3690624.3709179.

[36] Zeng, S., Zhang, J., He, P., Xing, Y., Liu, Y., Xu, H., Ren, J., Wang, S., Yin, D., Chang, Y., & Tang, J. (2024). The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG). *Findings of the Association for Computational Linguistics: ACL 2024*, 4505–4524. doi: 10.18653/v1/2024.findings-acl.267.

[37] Zou, W., Geng, R., Wang, B., & Jia, J. (2025). PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. *34th USENIX Security Symposium (USENIX Security 25)*, 3827–3844.