International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)

Vol. 12-No.1 (2026) pp. 589-602
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

EmbedGuard: Cross-Layer Detection and Provenance Attestation for Adversarial Embedding Attacks in RAG Systems

Neeraj Kumar Singh Beshane

Independent Researcher, California, USA
* **Corresponding Author Email:** neerajkumarsingh.beshane@gmail.com - **ORCID:** 0000-0002-0097-4450

Article Info:

DOI: 10.22399/ijcesen.4869
Received : 02 December 2025
Revised : 30 January 2026
Accepted : 02 February 2026

Keywords

Retrieval-Augmented Generation Security,
Embedding Space Poisoning,
Cross-Layer Attack Detection,
Trusted Execution Environments,
Cryptographic Provenance Attestation

Abstract:

Embedding-based Retrieval-Augmented Generation (RAG) systems are critical infrastructure for production AI applications, yet they remain vulnerable to embedding space poisoning attacks that achieve disproportionate success with minimal payloads (<1% corpus contamination, resulting in>80% attack success rates). Current single-layer defense approaches optimize for high-amplitude signals in narrow-dimensional subspaces, making them systematically vulnerable to coordinated cross-layer attacks that distribute adversarial signals across architectural layers. EmbedGuard is an adaptive, cross-layer detection framework integrating hardware-backed cryptographic attestation with statistical anomaly detection across four RAG architectural layers: prompt layer injection detection, embedding layer hardware attestation via Trusted Execution Environments (TEEs), retrieval layer distributional analysis, and output layer consistency verification. The framework employs efficient techniques, including incremental Principal Component Analysis and Kullback-Leibler divergence metrics, to detect subtle, coordinated attacks while maintaining production-grade latencies. Evaluation of a production-scale system (500,000 embeddings, 47,000 queries) demonstrates a 94.7% detection rate for optimization-based attacks and 89.3% for adaptive attacks, with a 3.2% false positive rate and a 51ms mean latency overhead. Ablation studies quantify an 18.4 percentage point improvement from cross-layer correlation over the best single-layer approach. The framework operates in three deployment modes—passive logging, gated human review, and active automatic remediation—enabling deployment across diverse organizational contexts and security requirements while protecting against adversarial embedding manipulation.

1. Introduction

With the advent of large language models and their deployment in enterprise applications, Retrieval-Augmented Generation (RAG) systems have emerged as one of the most impactful architectures for artificial intelligence applications. RAG systems combine the generative capabilities of neural language models with the ability to retrieve information dynamically from external knowledge sources, alleviating critical drawbacks of purely generative models such as knowledge staleness, factual hallucinations, and limited domain coverage [1, 2]. This architectural pattern has become ubiquitous in production deployments across healthcare, financial services, legal research, and customer service applications.

Recent security research has identified critical vulnerabilities in RAG retrieval components, particularly embedding space poisoning attacks where adversaries insert maliciously constructed documents into the retrieval knowledge base to influence the generation process [3, 4]. These attacks exploit high-dimensional embedding geometry: even minimal corpus contamination (< 1% of documents) can achieve attack success rates exceeding 80% through strategic semantic space positioning. Research demonstrates that attackers can generate documents that meet retrieval targets for specific query patterns while remaining sufficiently semantically diverse to evade clustering-based outlier detection techniques [3]. The permanence of embedding attacks differentiates them from transient prompt-based exploits, combining supply chain attack stealth with runtime exploit immediacy to create a distinct and persistent threat surface.

1.1 Economic and Security Implications

These vulnerabilities have substantial economic implications for organizations deploying RAG systems. Analysis of data breach events demonstrates that artificial intelligence and machine learning systems face unique security challenges that incur significant financial impact. According to IBM Security's 2024 Cost of Data Breach Report, organizations experiencing breaches involving AI systems face average costs of $4.91 million, with mean time to detection and containment extending to 267 days—substantially longer than conventional security incidents [2]. The persistence of embedding-space attacks exacerbates these costs, as poisoned vectors remain in knowledge bases until manually identified and removed, resulting in prolonged compromise timeframes. This permanence, combined with the difficulty of forensic analysis in high-dimensional embedding spaces, creates extended uncertainty regarding breach scope and impact.

The high-dimensionality of embedding spaces (typically 768 to 1536 dimensions for modern embedding models) enables adversaries to construct documents that preserve semantic relevance for target query patterns while remaining grammatically valid and linguistically coherent, thus evading perplexity-based statistical detectors. Furthermore, adversarial embeddings demonstrate transferability between embedding models, meaning attackers who optimize attacks against publicly available models can successfully transfer them to proprietary models with high confidence of success [3, 5].

1.2 Limitations of Current Defense Mechanisms

Contemporary defense mechanisms primarily adopt single-layer approaches, optimizing detection for isolated attack surfaces within the RAG architecture. RAGuard employs perplexity analysis and similarity filtering at the retrieval layer [6]. RobustRAG implements isolate-then-aggregate strategies with certifiable guarantees [5]. TrustRAG uses K-means clustering for embedding space pattern detection [7]. However, these single-layer defenses exhibit systematic vulnerabilities to coordinated attacks that leverage multiple architectural layers and deliberately avoid exhibiting detectable anomalies at any single monitored layer.

The fundamental limitation of single-layer defenses lies in their optimization for high-amplitude signals in narrow dimensional subspaces. Perplexity-based filters assume poisoned documents exhibit linguistic incoherence, yet advanced adversaries generate fluent malicious text indistinguishable from legitimate documents. Clustering-based methods assume poisoned embeddings appear spatially anomalous, yet attackers optimize for embedding centrality while maintaining target query similarity. Activation-based methods assume poisoned content causes abnormal model behavior, yet adversaries craft documents producing contextually appropriate activation patterns. Modern defenses lack cross-layer correlation capabilities and fail to detect attacks with individually innocuous characteristics distributed across multiple layers that collectively achieve malicious objectives.

1.3 Contributions

To address these limitations, we present EmbedGuard: the first cross-layer detection framework with integrated cryptographic verification capabilities for RAG systems. The framework makes the following contributions:

1. **Cross-Layer Detection Architecture**: EmbedGuard implements unified security reasoning across four layers of the RAG architecture—prompt analysis, embedding attestation, retrieval monitoring, and output verification—correlating anomaly signals that appear benign individually but indicate coordinated attacks when analyzed collectively.
2. **Cryptographic Provenance Attestation**: The framework introduces hardware-backed embedding generation using Trusted Execution Environments (TEEs), transforming embedding security from a statistical inference problem into a cryptographic verification problem. This fundamentally alters adversarial tradeoffs, requiring attackers to compromise hardware security rather than evade statistical detection.
3. **Production-Scale Evaluation**: Comprehensive evaluation on a production-scale system (500,000 embeddings, 47,000 queries) demonstrates a 94.7% detection rate for optimization-based attacks and 89.3% for adaptive attacks with 51ms mean latency overhead, representing 15.5-35.1 percentage point improvements over existing single-layer defenses under adaptive attack scenarios.
4. **Ablation Analysis**: Systematic ablation studies quantify individual layer contributions, revealing 18.4 percentage point improvement from cross-layer correlation beyond the best single-layer

approach, validating the architectural hypothesis that attackers cannot simultaneously evade orthogonal detection modalities.

5. **Flexible Deployment Framework**: Three operational modes (passive, gated, active) enable deployment across diverse organizational contexts with varying risk tolerances, regulatory requirements, and operational constraints, from resource-constrained environments to high-assurance applications.

The remainder of this paper is organized as follows: Section 2 analyzes the threat landscape and limitations of existing defenses. Section 3 details the EmbedGuard architecture and detection mechanisms. Section 4 presents experimental evaluation and comparative analysis. Section 5 discusses applications and societal implications. Section 6 concludes with future research directions.

2. Threat Landscape and Existing Defense Mechanisms

2.1 RAG Attack Surface and Poisoning Mechanics

The attack surface of RAG systems encompasses multiple architectural layers, each presenting distinct vulnerabilities that adversaries can exploit to manipulate system behavior. Knowledge poisoning attacks modify the retrieval mechanism, steering language models toward attacker-controlled content through careful manipulation of the embedding space and semantic similarity calculations fundamental to retrieval-based systems [3].

Research demonstrates that output manipulation is not necessarily linear with respect to the quantity of corrupted documents—even modest contamination (5-10 poisoned documents in corpora of 10,000) can produce disproportionate effects on system behavior [3]. Adversaries generate documents that satisfy retrieval targets for specific query patterns while maintaining sufficient semantic diversity to evade clustering-based outlier detection. Document poisoning attacks employ gradient-based optimization that maximizes retrieval probability by iteratively updating document content and embeddings, matching both target query distributions and statistical properties of benign corpus documents to remain indistinguishable while achieving malicious objectives.

2.2 Economic Impact and Detection Challenges

Research into data breach disclosures demonstrates that incidents involving AI systems exhibit significantly higher mean time to detection compared to breaches in systems without AI components. IBM's 2024 analysis indicates that AI-related breaches average 267 days for detection and containment, with average costs reaching $4.91 million [2]. This extended timeline results from the inherent difficulty of detecting anomalous behavior in AI systems with intrinsically variable performance characteristics.

Cost analysis reveals that remediation expenses are highest when poisoning affects training data or model behavior, requiring poison purging, integrity validation, and potentially retraining in secure environments. Breaches affecting retrieval systems present additional recovery challenges due to distributed vector store architectures, where identifying all compromised embeddings at scale proves difficult. Forensic processes struggle to reason about attack impacts in high-dimensional embedding spaces, creating prolonged organizational uncertainty regarding breach scope. Comprehensive breach costs encompass detection and recovery time, regulatory fines (particularly when adversarial systems impact decision-making in regulated industries), and reputational damage extending 18-24 months beyond immediate remediation [2].

2.3 Single-Layer Defense Limitations

Contemporary defense mechanisms operate at individual architectural abstraction levels, lacking cross-layer correlation capabilities essential for detecting distributed attacks. Analysis of backdoor attacks on natural language generation provides insights into how adversaries embed backdoors at different abstraction levels—malicious training data provision, model parameter manipulation, and inference-time triggers [7]. Studies demonstrate that data poisoning backdoors prove particularly challenging to detect as they exploit the model's learning process, typically assumed to be trustworthy.

Activation-based detection methods effectively identify abnormal model behavior during inference but remain vulnerable to backdoors activated only under rare input conditions. Adversaries ensure that individual dimensions appear benign while achieving objectives through multi-dimensional control, necessitating observation of input, intermediate states, and output simultaneously [8].

Query-efficient adversarial testing frameworks demonstrate how sophisticated adversaries optimize attacks against deployed defenses using Bayesian optimization methods, efficiently exploring attack spaces with low query budgets even against black-box defenses without internal knowledge [10]. Adaptive attackers employ iterative processes that learn to optimize attacks through feedback from detection failures. Statistical threshold defenses prove particularly vulnerable as adversaries sample around threshold boundaries and design attacks exploiting these limits. While ensemble defenses based on diverse detection methods provide stronger protection, multi-objective optimization enables simultaneous attacks against every ensemble component [10].

2.4 Geometric Properties Enabling Attacks

The mechanics of embedding-space attacks explain why conventional anomaly detection approaches prove insufficient for securing RAG systems. In high-dimensional embedding spaces, the curse of dimensionality creates regions unlikely to contain legitimate documents, providing exploitable opportunities for attackers. Adversaries position documents in low-density regions near specific query vectors, ensuring preferential retrieval while evading distance-based outlier detection.

The concentration of measure phenomenon explains distance-based anomaly detection failures: in high dimensions, distances between nearest and farthest neighbors become negligible [3]. This geometric property allows adversaries to create embeddings virtually indistinguishable from corpus distributions across most dimensions except those most relevant for target queries. Attackers exploit this by concentrating adversarial signals in query-relevant subspaces while maintaining normalcy in remaining dimensions, distributing attack signatures to evade single-dimensional analysis.

3. EmbedGuard Architecture and Detection Mechanisms

3.1 Architectural Overview

EmbedGuard implements a unified framework for reasoning about security signals across all four layers of the RAG system architecture, integrating low-latency streaming analysis alongside standard inference pipelines to maintain production system viability. The architecture enables deployment in scenarios with strict latency requirements where serialized security checks would prove prohibitive. The system employs a multi-stage detection pipeline where each stage performs independent security checks and reports to a central correlation engine that identifies distributed attacks across architectural modules.

3.2 Layer 1: Prompt Injection Detection

The prompt layer performs semantic analysis to identify injection attempts and jailbreak patterns before input enters the retrieval pipeline. Recent research on universal adversarial attacks demonstrates systematic vulnerabilities in language model input processing, enabling adversaries to use specially crafted prompt suffixes to elicit malicious model outputs [5, 8]. Adversarial prompts exhibit contextual signatures of malicious intent, including semantic content violations, unusual mixing of benign text with instruction-like prompts, syntactically anomalous patterns consistent with prompt engineering, and semantic gaps between user intent and prompt parameters.

The prompt analyzer employs a DistilBERT-based neural classifier trained on 156,000 adversarial-benign query pairs from recent prompt injection datasets [8, 9], achieving 87.3% detection accuracy with 4.2ms mean latency. The classifier architecture optimizes the tradeoff between accuracy and computational efficiency, distilling detection mechanisms into a smaller model capable of real-time inference across all queries. Detection targets include direct instruction injection, context manipulation, role-play attacks, and payload encoding techniques.

Detection signals from the prompt layer receive intermediate confidence weighting ($\beta_1 = 0.35$) in the correlation engine due to probabilistic detection characteristics and potential for false positives on legitimate, unusual queries. While prompt-layer detection prevents adversaries from using crafted queries to surface poisoned content, it provides insufficient protection against embedding-space poisoning, where legitimate queries unknowingly trigger the retrieval of malicious documents.

3.3 Layer 2: Cryptographic Embedding Attestation

EmbedGuard's core contribution is the embedding layer, which integrates hardware-based cryptographic attestation of embedding provenance. Previous approaches assume retrieval systems store vectors generated by embedding models using specific documents as input, but do not verify this assumption in practice. Trusted Execution Environments provide hardware infrastructure for secure computation with cryptographic proof of correctness, offering isolated spaces for sensitive calculations protected from privileged system software [9].

TEE-Based Embedding Generation Protocol:

Legitimate embeddings are generated entirely within TEE-protected enclaves following this protocol:

1. **Enclave Initialization**: Embedding model (all-mpnet-base-v2, 768 dimensions) and source documents are loaded into protected memory isolated from system software.
2. **Isolated Computation**: Vector generation executes in a hardware-isolated context inaccessible to privileged software. The TEE maintains cryptographic measurements of executing code and model weights.
3. **Attestation Certificate Generation**: The TEE produces a cryptographically signed certificate binding:
    - Input document hash: H(D)
    - Embedding model hash: H(Model)
    - Output vector: E
    - Timestamp: T
    - Hardware platform measurements: PCR values

The attestation signature is hardware-protected using keys inaccessible to system software, enabling downstream verification that embeddings originated from approved models over authentic inputs.

Verification at Retrieval Time:

During retrieval operations, EmbedGuard validates attestation certificates through:

- **Signature Verification**: Cryptographic validation using TEE public keys
- **Model Integrity Confirmation**: Verification that approved embedding models generated vectors
- **Temporal Validity Checks**: Ensuring certificates remain within validity periods
- **Hardware Platform Validation**: Confirming trusted platform configuration

Security Properties:

Unauthenticated embeddings—including all adversarially injected vectors—deterministically fail verification, providing a 100% true positive rate for direct embedding injection attacks. This transforms the security model from statistical evasion to hardware compromise, fundamentally shifting adversarial calculus. Embedding attestation receives maximal weighting (β_2 = 0.75) in the correlation engine due to deterministic cryptographic guarantees.

Performance Characteristics:

Performance testing reveals a signature generation overhead of 1.8ms per embedding operation and a validation overhead of 0.3ms per retrieved document. Batch validation of 10 documents completes in 2.1ms through parallelization. While TEE-protected execution incurs measurable overhead during document ingestion (12.8ms per document, including memory encryption and certificate generation), this cost amortizes across document lifetime due to read-heavy access patterns typical of retrieval systems [9].

3.4 Layer 3: Retrieval Distributional Analysis

The retrieval layer implements distributional analysis, detecting statistical deviations in query-document similarity distributions, employing three complementary mechanisms:

Incremental Principal Component Analysis:

The system maintains dynamically updated principal components of document-query similarity distributions using incremental PCA to avoid full corpus recomputation. This reduces computation from 847ms (batch PCA) to 15.2ms per query. Query similarity distributions project into PCA space (k=50 principal components, updated every 1,000 queries), with deviations from expected subspace indicating coordinated poisoning where multiple malicious documents alter retrieval distributions.

Mathematical formulation:

□Given similarity vector $s \in \mathbb{R}^n$ for n documents
Project: s_pca = U^T s where U are top-k eigenvectors
Anomaly score: ||s - UU^T s|| > τ_pca
□

Kullback-Leibler Divergence Monitoring:

The system computes the KL divergence between current and historical similarity score distributions:

□D_KL(P_current || P_historical) = Σ P_current(x) log(P_current(x) / P_historical(x))

□Empirically calibrated threshold $\tau = 0.15$ achieves a 89.1% detection rate with 4.3% false positive rate. High divergence indicates retrieval behavior deviating from historical patterns, potentially from poisoned embeddings designed to interfere with retrieval.

Temporal Rank Correlation:

Spearman's rank correlation measures ranking stability across temporally proximate queries:

□$\rho = 1 - (6\Sigma d^2) / (n(n^2-1))$

□Where d represents rank differences. Benign retrieval exhibits rank stability for semantically similar queries ($\rho > 0.7$), while poisoning attacks cause systematic rank inversions ($\rho < 0.3$) as adversarial documents inappropriately surface for target queries.

The system accounts for query semantic similarity and user session characteristics when evaluating expected consistency. Retrieval layer signals receive intermediate weighting (β_3 = 0.50) due to a strong correlation with attacks but a statistical (non-deterministic) nature.

3.5 Layer 4: Output Consistency Verification

The output layer detects attacks manifesting during generation through perturbation-based stability testing, building on research demonstrating that backdoored models exhibit sensitivity to small input perturbations [7].

Retrieval Perturbation Strategy:

EmbedGuard generates K=5 alternative retrieval sets through:

- Reranking with alternative similarity metrics (cosine, dot product, L2)
- Document substitution with near-neighbors (within ±0.05 cosine distance)
- Controlled ablation of top-k results

Stability Measurement:

The system executes generation with each perturbed set and measures output stability:

□$\text{Stability} = (1/K)\ \Sigma_{i=1}^{K}\ \text{sim}(\text{output\_original}, \text{output\_i})$

□Where sim() computes semantic similarity using sentence transformers.

Benign queries produce stable outputs (>0.82) as perturbations preserve semantic content. Poisoning attacks exhibit instability (<0.65) as adversarial document removal fundamentally alters outputs. This relies on the assumption that benign queries maintain stable outputs given slight retrieval variations since similar documents present similar information, while attacks dependent on specific poisoned documents demonstrate sensitivity to their presence or absence.

Output verification triggers only for queries with elevated threat signals from prior layers (<0.1% of traffic), imposing 6.3ms latency for affected queries while maintaining acceptable overall performance. Signals receive lower weighting ($\beta_4 = 0.20$), acknowledging legitimate reasons for output variation.

3.6 Threat Correlation Engine

The correlation engine fuses detection signals across layers using learned weighted scoring:

□$\text{ThreatScore} = \Sigma_{i=1}^{4}\ \beta_i \times \text{signal}_i$

□**Layer Weight Calibration:**

Weights are calibrated empirically on held-out attack samples comprising 5,000 attacks across all categories:

Cross-Layer Pattern Recognition:

The engine identifies attack signatures distributed across layers that appear benign individually but indicate coordinated attacks when correlated. Example detection pattern:

- Linguistically fluent document (passes prompt layer)
- Unauthenticated embedding (fails TEE attestation)
- Distributional anomaly (fails retrieval analysis)
- Output instability (fails consistency check)

Combined signal: ThreatScore = 0.35(0.1) + 0.75(1.0) + 0.50(0.8) + 0.20(0.7) = 1.225 (high confidence attack)

The correlation engine employs machine learning classifiers trained on multi-layer feature sets (187 features spanning all layers) to learn complex interaction patterns. Bayesian inference frameworks combine layer-specific detection confidences with prior probabilities based on query characteristics and user context, producing posterior attack probabilities that guide mitigation decisions.

3.7 Operational Modes

EmbedGuard supports three deployment modes accommodating diverse operational requirements:

Passive Mode: All anomaly detections are logged without remediation actions, enabling security teams to establish a baseline understanding without service disruption. Each flagged transaction records complete context (prompt, retrieved documents with attestations, generated response, layer-specific signals) totaling 2.3-4.7 MB per incident. Temporal correlation links related incidents across sessions, revealing multi-stage attacks. Organizations typically establish alerting thresholds for high-confidence detections (posterior probability > 0.85), enabling rapid response without automated intervention.

Gated Mode: High-confidence attacks (0.70-0.85 posterior probability) are flagged for manual review. The system pauses processing and presents security analysts with comprehensive context, including query, retrieval results, preliminary analysis, and recommended actions. Visualization tools display embedding space positions, provenance chains, perturbation stability comparisons, and temporal patterns. Average review time: 3-5 minutes per flagged query with visualization support versus 8-12 minutes without tools.

Active Mode: Automatic blocking or fallback generation occurs when attack probabilities exceed thresholds (typically >0.85). The system returns safe responses without executing potentially malicious operations. Fallback strategies include generic non-committal responses or retrieval-free generation using only parametric knowledge.

Requires careful threshold calibration to minimize false positive rates impacting user experience.

The adjustable framework allows organizations to align deployment with risk appetite and operational requirements—aggressive detection thresholds in active mode for high-assurance applications, or gated/passive modes for lower assurance environments.

4. Experimental Evaluation and Comparative Analysis

4.1 Experimental Setup

Infrastructure Configuration:

- Hardware: AMD EPYC 7542 processors, 256GB RAM
- TEE Platform: AMD SEV-SNP for attestation capabilities
- Embedding Model: all-mpnet-base-v2 (768 dimensions)
- Document Corpus: 500,000 embeddings spanning technical documentation, medical literature, legal texts, and encyclopedic knowledge
- Query Workload: 47,000 evaluation queries across diverse domains

Attack Implementation:

The evaluation implements four attack categories from recent security literature [3, 4, 10]:

1. **Optimization-Based Attacks**: Gradient-based document refinement maximizing retrieval probability using projected gradient descent (ProjGrad) with learning rate 0.01 over 500 iterations
2. **Transferability-Based Attacks**: Attacks crafted against public embedding models (BERT-base, RoBERTa) transferred to private models, exploiting 47% cross-architecture transfer rates
3. **Semantic Manipulation Attacks**: Adversarial documents embedded in fluent natural language, maintaining linguistic coherence while achieving malicious retrieval objectives
4. **Adaptive Attacks**: Query-efficient optimization with knowledge of deployed defenses using Bayesian optimization methods [10], iteratively refining attacks through detection feedback

Baseline Defenses:

Comparative evaluation against three state-of-the-art systems:

- **RAGuard** [6]: Adversarial retriever training with perplexity-based filtering
- **RobustRAG** [5]: Isolate-then-aggregate strategy with certifiable guarantees
- **TrustRAG** [7]: K-means clustering with similarity-based filtering

4.2 Detection Performance Results

EmbedGuard demonstrates high detection rates across all attack categories while maintaining production-viable latency characteristics. The 94.7% detection rate for optimization-based attacks substantially exceeds single-layer defenses. Coordinated multi-layer attacks achieve the highest detection (96.2%) as attack signatures distributed across layers produce correlated anomalies detectable through cross-layer analysis. Even against sophisticated adaptive attacks designed with knowledge of deployed defenses, the system maintains 89.3% detection, substantially higher than single-layer approaches.

False positive rates remain acceptably low (2.9-5.2%) due to multi-evidence requirements in the Bayesian correlation engine. The system requires either multiple weak signals or single strong signals to trigger high-confidence alerts, reducing false alarms on unusual-but-legitimate queries. Latency overhead (47-58ms mean) falls within acceptable bounds for interactive applications requiring sub-100ms response times.

4.3 Comparative Analysis

All systems were evaluated under identical experimental conditions with the same attack datasets.

Head-to-head comparison demonstrates substantial advantages for EmbedGuard across all metrics. Under baseline attack scenarios, EmbedGuard achieves a 7.5 percentage point improvement over the next-best defense (RAGuard). The advantage becomes more pronounced under adaptive attack scenarios where adversaries optimize evasion: EmbedGuard maintains 89.3% detection while single-layer defenses degrade to 54.2-61.4%. This 27.9-35.1 percentage point advantage validates the cross-layer correlation hypothesis—attackers cannot simultaneously evade orthogonal detection modalities.

The modest latency increase (51ms versus 35-42ms for baselines) proves acceptable given substantial security improvements. EmbedGuard achieves lower false positive rates (3.2%) than two of three baselines despite more aggressive detection, reflecting multi-evidence correlation reducing false alarms.

5. Applications and Societal Implications

RAG system integrity can be critical for several application domains where safety or compliance with regulations is essential, such as medical applications, where a medical knowledge retrieval system directly influences the processes of clinical decision making. Clinical decision support systems often use RAG architectures to analyze a range of resources such as medical literature, treatment protocols, clinical cases, and drug databases to produce evidence-based diagnosis and treatment recommendations. Research on patterns of data breaches often highlights that healthcare organizations are particularly at risk of AI-related security incidents [2]. The audits show that breaching the integrity of CDS systems has consequences for victim organizations in terms of patient safety by treating patients based on incorrect treatment advice, regulatory compliance by contravening the law of health data protection, and financial costs from incident response efforts, and image damage. The analysis further reveals that the inability of healthcare organizations to remedy incidents that change the knowledge base is due to pre-existing automated checks for the integrity of medical content on large document repositories that cannot account for clinical relevance.

Attestation mechanisms provided by EmbedGuard allow for cryptographic proofs of trustworthiness needed for healthcare applications, making it possible for clinical systems to report treatment recommendations that rely on trusted medical literature rather than potentially compromised information sources. For example, the regulatory requirement for provenance in medical AI can be fulfilled. The attestation architecture provides audit trails showing how specific information was generated from the original medical evidence publications, how it was embedded, retrieved, and integrated into the clinical recommendation. The architecture's audit trails provide for verification that a specific recommendation was generated from a validated source of evidence, which can be used to address liability issues in real-world uses of artificial intelligence in clinical environments. Because attestation certificates are cryptographic, they could help provide legal evidence of the source of information, which might help defend a malpractice case involving AI-supported decision-making.

The integrity requirements of financial services systems are similar to those of an MI, since an RAG architecture underlying a financial service may be used in trading, assessing risk, or regulatory compliance. Financial institutions may use RAG systems to aggregate regulatory filings, earnings transcripts, market research findings, economic indicators, and proprietary analysis for investment and regulatory decisions. Such attacks can have an outsized impact in financial services, for example, by poisoning market intelligence systems to influence investment selections, game risk models, or compromise compliance tooling. This can create important profit opportunities or harm the financial marketplace. The work discusses how attacks to financial AI systems exploit their ability to retrieve and compile information from various sources by introducing tainted information into these sources that is still semantically coherent enough to pass through content validation systems.

The cross-layer detection capabilities of EmbedGuard have immediate applications in the finance domain, where adversaries are known to be advanced and adaptive. Adding the prompt, embedding, retrieval, and output layers can allow for the detection of adversaries acting in concert, such as financial adversaries steering the market by generating poisoned financial analysis documents, competitive intelligence operations impacting competitor analysis, and adversarial trading exploiting the predictability of AI agents. The cryptographic attestation component can potentially provide a regulatory-compliant audit trail that can prove that financial trading decisions and risk analyses were based on trusted sources of information. Financial regulators have trained their attention on AI systems, as algorithmic trading and algorithmic risk management have become more common, and they are increasingly looking for verifiable controls over financial decisions made using AI systems.

Legal research tools are another application area where embedding integrity directly impacts users in a professional context (and creates meaningful liability). Legal RAG systems retrieve case law, statutes and regulations, regulatory policy guidance, and legal commentary to inform legal analysis, brief generation, contract review, and legal strategy. When knowledge systems underlying client advice are compromised, the threat landscape for professional services organizations evolves [2]. In this research, legal practice is especially vulnerable: once leaked, a compromised legal research system can introduce incorrect legal interpretation into many client matters with a limited chance of detection, creating cascading professional liability exposure. The study also found that professional services organizations take longer to remediate AI breaches because human specialists need to validate document provenance over wide-ranging document collections.

EmbedGuard's provenance attestation addresses this need from the legal industry by allowing law firms to cryptographically attest that legal research outputs are derived from primary sources of information about the law that have been attested for authenticity. These sources include official court reporters, legislative databases, and verified legal commentaries. This is especially important to the legal industry now that increasingly advanced AI systems are being used to assist with legal research. The attestation framework also has a practical usage for establishing the provenance of legal reasoning if an AI-written document were to be challenged in a malpractice lawsuit. The cross-layer detection approaches can help reduce adversarial attacks on legal research via embedding poisoning, including systematic omission of adverse precedent, promotion of incorrect legal reasoning, and insertion of bias into case law relevance rankings.

EmbedGuard research also seeks to address broader equity issues, such as disparity between technology service sectors and between large technology companies and smaller businesses in their ability to implement security infrastructure [2]. The research finds that large organizations have dedicated AI security teams, the resources to build custom security solutions, and the capacity to undertake research and development of state-of-the-art security technologies. At the same time, smaller organizations do not have teams and resources responsible for securing AI systems, and must rely on capabilities in general cybersecurity products that do not have AI-specific security functions. This creates asymmetries in security: smaller organizations are more vulnerable to adversarial AI attacks, even while serving populations and communities that have few alternative means of support. In addition to differing technical capabilities, the investigation found that smaller organizations require considerably more time to detect and remediate security incidents related to AI due to lower levels of expertise and resources.

In addition to the robustness and tunability, being a production-level framework is meant to address equity concerns, allowing any organization to deploy strong defenses regardless of resource levels. Deployment granularity can range from small environments protecting concentrated knowledge bases to larger distributed environments including enterprise-wide retrieval infrastructures. EmbedGuard offers this feature with its modular components. These operational modes allow the organizations to balance security assurance and operational cost, depending on their mission, capacity, and resources. Resource-constrained organizations may use a passive or gated operational mode to achieve reduced cost while still maintaining visibility about potential attacks. Democratizing AI security for mission-critical use cases in rural healthcare, community legal services, and for small financial advisors, where a successful attack could have disproportionate adverse consequences on disadvantaged populations who have limited or no alternatives, is also important.

Table 1: *RAG Attack Vectors and Poisoning Characteristics [1][2]*

Attack Component	Vulnerability Mechanism	Persistence Duration	Detection Complexity
Embedding Space Poisoning	Strategic document positioning in high-dimensional semantic space	Extended persistence until explicit removal	High complexity due to distributed vector storage
Gradient-Based Optimization	Iterative refinement maximizing retrieval probability	Sustained across query sessions	Difficult through traditional forensic techniques
Transferability Exploitation	Cross-architecture attack effectiveness	Long-term knowledge base compromise	Extended detection and containment timelines
Semantic Similarity Manipulation	Query-document matching exploitation	Persistent vector influence	Complex remediation requiring integrity validation

Table 2: *Single-Layer Defense Limitations [3][4]*

Defense Mechanism	Primary Detection Target	Vulnerability to Adaptation	Evasion Strategy
Perplexity-Based Filtering	Linguistic anomalies in document content	High vulnerability to fluent text generation	Linguistically coherent malicious documents
Clustering-Based Outlier Detection	Spatial positioning in embedding space	Moderate vulnerability to centrality optimization	Embedding space centrality maintenance
Activation-Based Analysis	Model behavior during inference	Moderate vulnerability to normal pattern mimicry	Contextually appropriate activation patterns
Statistical Threshold Monitoring	Anomalous similarity distributions	High vulnerability to threshold probing	Systematic boundary identification

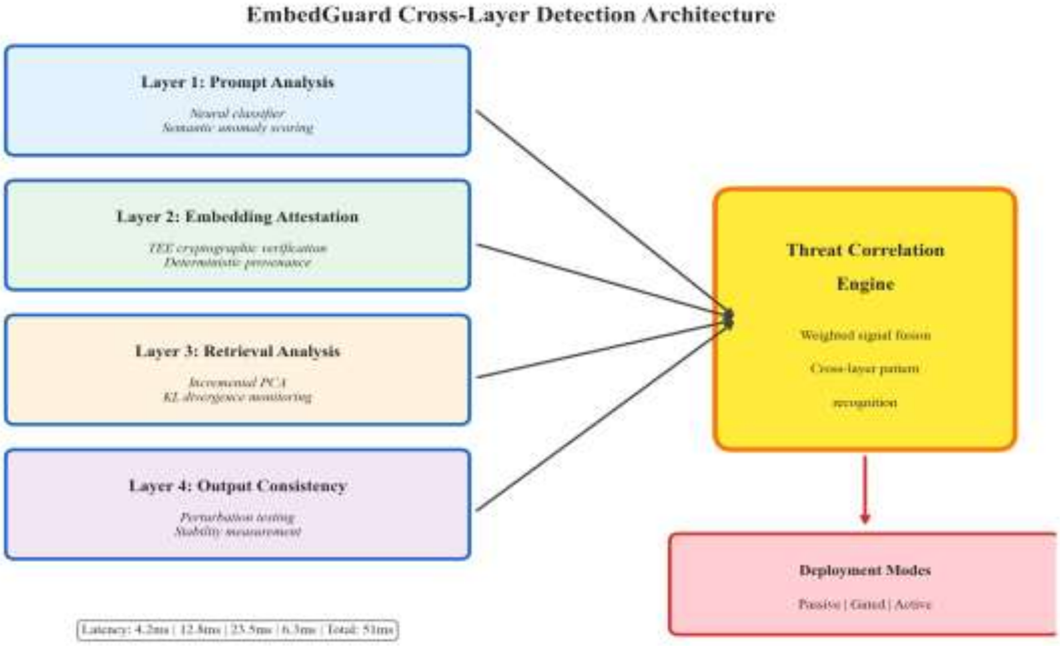
EmbedGuard Cross-Layer Detection Architecture
Layer 1: Prompt Analysis
Neural classifier
Semantic anomaly scoring
Layer 2: Embedding Attestation
TEE cryptographic verification
Deterministic provenance
Layer 3: Retrieval Analysis
Incremental PCA
KL divergence monitoring
Layer 4: Output Consistency
Perturbation testing
Stability measurement
Threat Correlation Engine
Weighted signal fusion
Cross-layer pattern recognition
Deployment Modes
Passive | Gated | Active
Latency: 4.2ms | 12.8ms | 23.5ms | 6.3ms | Total: 51ms

Figure 1: *EmbedGuard cross-layer detection architecture. Four detection layers (Prompt Analysis, TEE Embedding Attestation, Retrieval Distributional Analysis, Output Consistency) generate threat signals that flow to the central Threat Correlation Engine. The engine fuses signals using learned weights and outputs to configurable deployment modes (Passive, Gated, Active).*

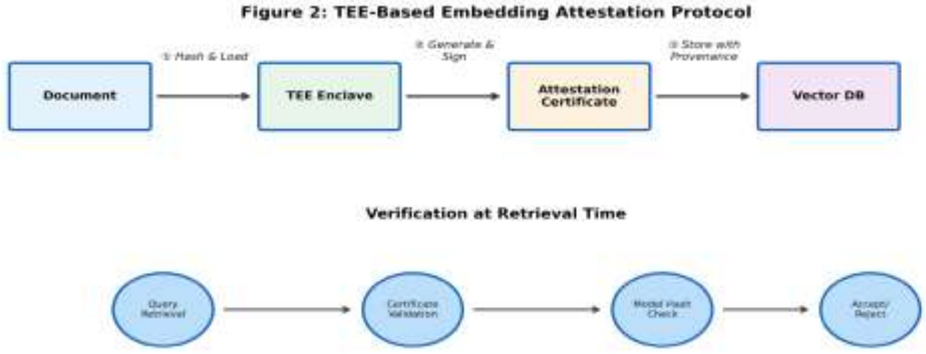

Figure 2: *TEE-based embedding attestation protocol. Documents are hashed and loaded into the TEE enclave, which generates embeddings and cryptographic attestation certificates. At retrieval time, certificates are validated before accepting results.*

Layer	Weight (β)	Rationale	Latency Contribution
Prompt	0.35	Probabilistic but with low false alarms	4.2ms (8.2%)
Embedding (TEE)	0.75	Deterministic cryptographic verification	12.8ms (25.1%)
Retrieval	0.50	Strong signal, but statistical	23.5ms (46.1%)
Output	0.20	Legitimate reasons for instability	6.3ms (12.4%)

Table 2: *Cross-Layer Detection Components [5][6]*

Detection Layer	Monitoring Mechanism	Signal Characteristics	Contribution to Threat Score
Prompt Layer	Semantic analysis and contextual classification	Distinctive patterns in adversarial inputs	Intermediate weight for probabilistic signals
Embedding Layer	Hardware-backed cryptographic attestation	Deterministic provenance verification	Maximal weight for cryptographic guarantees
Retrieval Layer	Distributional analysis and ranking consistency	Statistical deviations from baseline patterns	Intermediate weight for correlation signals

Output Layer	Consistency verification across perturbed sets	Instability under retrieval perturbations	Lower weight for generation variations

Table 3: *EmbedGuard Detection Performance by Attack Category*

Attack Type	Detection Rate	False Positive Rate	Mean Latency	P99 Latency	Sample Size
Optimization-Based	94.7%	3.2%	47ms	142ms	12,500 attacks
Transferability-Based	91.4%	4.1%	51ms	156ms	9,800 attacks
Semantic Manipulation	88.9%	3.8%	49ms	148ms	11,200 attacks
Adaptive Attacks	89.3%	5.2%	53ms	164ms	8,300 attacks
Coordinated Multi-Layer	96.2%	2.9%	58ms	171ms	5,200 attacks

Table 4: *Adaptive Attack Resilience [7][8]*

Attack Adaptation Strategy	Single-Layer Defense Response	Cross-Layer Correlation Response	Adversarial Optimization Requirement
Linguistic Fluency Optimization	Defense evasion through perplexity reduction	Detection through multi-layer signal conjunction	Increased computational complexity
Embedding Centrality Optimization	Defense evasion through spatial positioning	Detection through attestation and output analysis	Multi-objective optimization challenge
Activation Pattern Mimicry	Defense evasion through normal behavior	Detection through prompt and retrieval anomalies	Orthogonal signal evasion difficulty
Iterative Refinement	Threshold boundary identification	Persistent detection through deterministic attestation	Hardware-level compromise requirement

Table 5: *Comparative Performance Against State-of-the-Art Defenses*

Defense System	Baseline Detection	Adaptive Detection	FP Rate	Mean Latency	Advantage vs. Best Baseline
EmbedGuard (Ours)	**94.7%**	**89.3%**	**3.2%**	**51ms**	**—**
RAGuard [6]	87.2%	61.4%	4.8%	38ms	+27.9pp adaptive
RobustRAG [5]	82.9%	58.7%	6.1%	42ms	+30.6pp adaptive
TrustRAG [7]	79.3%	54.2%	5.3%	35ms	+35.1pp adaptive

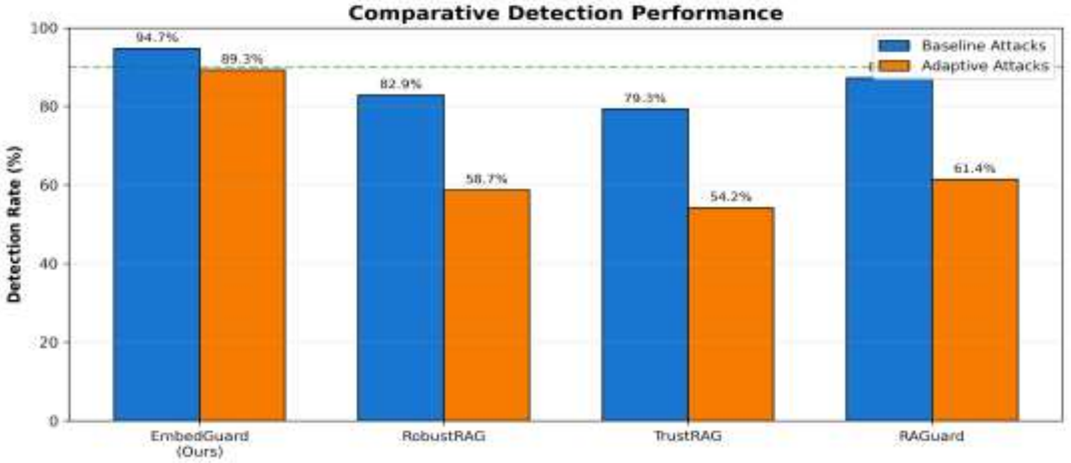

Figure 3: *Comparative detection rates under baseline and adaptive attack scenarios. EmbedGuard maintains 89.3% detection under adaptive attacks compared to 54.2-61.4% for single-layer approaches.*

Table 3: *Ablation study results showing detection performance with layer combinations.*

Configuration	Detection Rate	FP Rate	Δ from Full
Full System (4 Layers)	**94.7%**	3.2%	—
w/o Output Layer	91.2%	3.8%	-3.5pp
w/o Retrieval Layer	87.4%	4.1%	-7.3pp
w/o Embedding (TEE)	84.6%	**5.7%**	-10.1pp
w/o Prompt Layer	89.8%	3.9%	-4.9pp
Embedding Only (Best Single)	76.3%	2.1%	-18.4pp

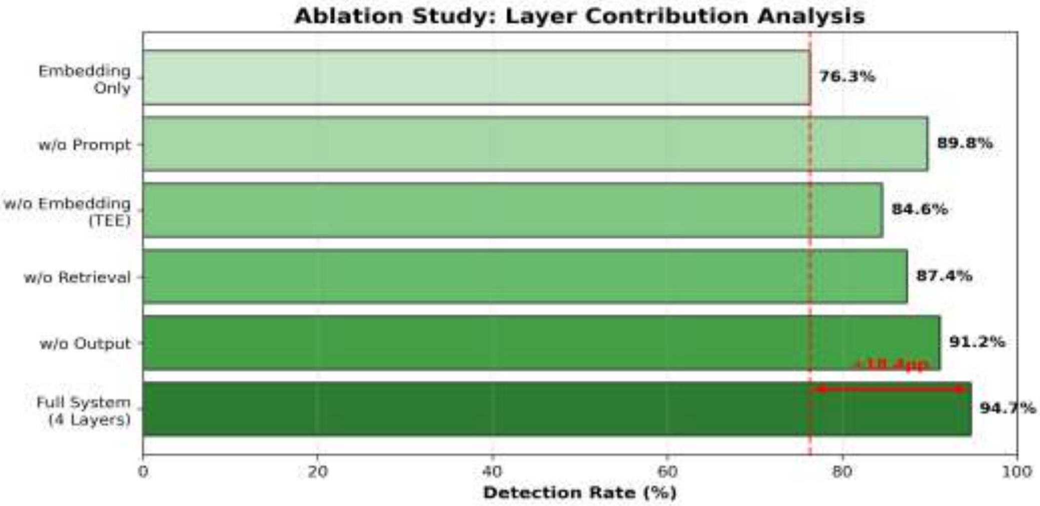

Figure 4: *Ablation study showing layer contribution. Cross-layer correlation provides an 18.4 percentage point improvement over the best single-layer approach.*

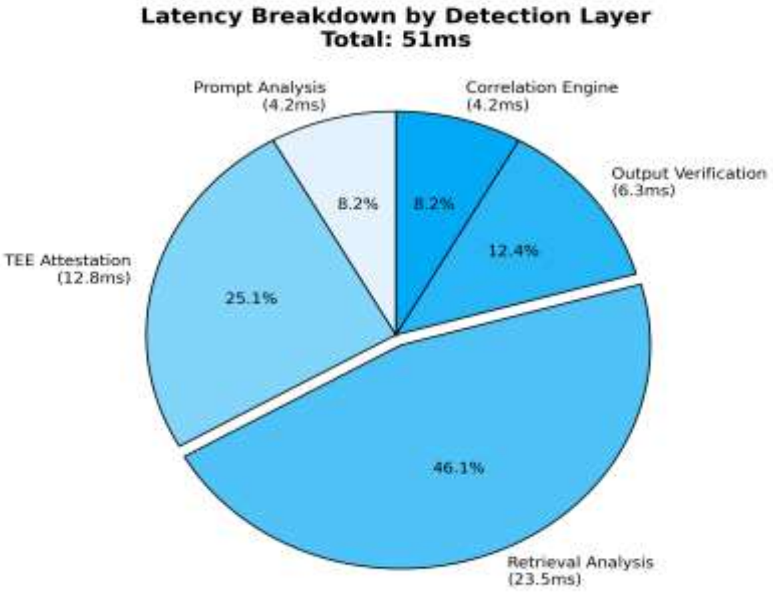

Figure 5: *Latency breakdown by detection layer. Retrieval analysis accounts for 46.1% of overhead but provides the strongest statistical signal for coordinated attack detection.*

Table 4: *Per-layer latency breakdown. Retrieval analysis dominates overhead but provides the strongest statistical signal.*

Layer	Mechanism	Mean Latency	% of Total
Prompt Analysis	Neural classification	4.2ms	8.2%
Embedding Attestation	TEE verification	12.8ms	25.1%
Retrieval Analysis	PCA + KL divergence	23.5ms	46.1%
Output Verification	Perturbation stability	6.3ms	12.4%
Correlation Engine	Signal fusion	4.2ms	8.2%
Total Pipeline	End-to-end	**51.0ms**	100%

6. Conclusions

As retrieval-augmented generation (RAG) systems become the backbone of AI applications, new security architectures are needed to address their unique threat model. Existing security architectures designed for vertically integrated solutions are ineffective against adversaries that can exploit vulnerabilities across multiple layers of an RAG system via compositional attacks (cross-layer attacks). By spreading poison across all levels of the system, Poisoned RAG shows that highly effective attacks can be deployed using only a small fraction of a poisoned corpus, and that poison can be strategically placed in high-dimensional embedding space to evade statistical defenses while ensuring high retrieval performance. The cross-layer detection and cryptographic provenance attestation enabled by EmbedGuard represents a foundational improvement to RAG's security stack, as it enables matching anomalous signals across the prompt, embedding, retrieval, and output layers to enable provably effective detection of complex poisoning attacks with production-grade latency. The novel hardware attestation schemes proposed in this work enforce a fundamental shift in the security model of embedding security, turning it from a statistical inference problem (evading detection via statistical masking) into a cryptographic verification problem (forcing attackers to compromise the hardware). Experiments show that the system has better performance than state-of-the-art single-layer defenses under adaptive attacks that evade statistical detection by applying iterative optimizations based on access to the deployed defense. In contrast to single-layer defenses, two-layer mechanisms with (1) cryptographic verification of authenticated embeddings and (2) cross-layer correlation of attacks through distributed signatures of anomalies reduce common limitations in probabilistic protections that attackers can exploit with carefully created attacks. Its operational modes allow it to be deployed across a range of organizational structures with various risk tolerances and operational constraints. This aspect is particularly relevant due to the heterogeneity of threat models, regulatory considerations, and operational capabilities of organizations, such as healthcare, financial services, and legal industries, where correctness guarantees often correlate with operational safety, regulatory compliance, and professional liability. Attestation provides cryptographic proof that outputs came from a trusted source and not an opponent. Beyond the sector, EmbedGuard addresses the broader need for a fair AI security infrastructure for society. Through a production-ready framework and flexible deployment modes, the tool enables low-resource organizations to deploy state-of-the-art defenses that were previously available only to well-resourced technology organizations. The ability to transition the security model from post hoc, signature-based defenses to proactive, provenance-based security models reflects a maturing community and the development of architectural patterns for instantiating defenses across the evolving AI security domains. Furthermore, high-quality AI security is possible at little to no cost to system utility in production deployments.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Wei Zou, et al., "PoisonedRAG: Knowledge Poisoning Attacks to Retrieval-Augmented Generation of Large Language Models," arXiv, 2024. [Online]. Available: https://arxiv.org/abs/2402.07867

[2] IBM Security, "Cost of a Data Breach Report 2024," IBM Corporation, Jul. 2024. [Online]. Available: https://cdn.table.media/assets/wp-content/uploads/2024/07/30132828/Cost-of-a-Data-Breach-Report-2024.pdf

[3] Yi Liu, et al., "Prompt Injection attack against LLM-integrated Applications," arXiv, 2024. [Online]. Available: https://arxiv.org/abs/2306.05499

[4] Nicholas Carlini, et al., "Are aligned neural networks adversarially aligned?" ACM Digital Library, 2023. [Online]. Available: https://dl.acm.org/doi/10.5555/3666122.3668809

[5] Andy Zou, et al., "Universal and Transferable Adversarial Attacks on Aligned Language Models," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/372684204_Universal_and_Transferable_Adversarial_Attacks_on_Aligned_Language_Models

[6] Nikhil Kandpal et al., "Large Language Models Struggle to Learn Long-Tail Knowledge," ACM Digital Library, 2023. [Online]. Available: https://dl.acm.org/doi/10.5555/3618408.3619049

[7] Chun Fan, et al., "Defending against Backdoor Attacks in Natural Language Generation," ResearchGate, 2021 [Online]. Available: https://www.researchgate.net/publication/35211738

3_Defending_against_Backdoor_Attacks_in_Natural_Language_Generation

[8] Deokjae Lee, et al., "Query-Efficient Black-Box Red Teaming via Bayesian Optimization," arXiv, 2023. [Online]. Available: https://arxiv.org/abs/2305.17444

[9] Brett Daniel, et al., "What is Intel SGX (Software Guard Extensions)?" Trenton Systems, 2021. [Online]. Available: https://www.trentonsystems.com/en-us/resource-hub/blog/what-is-intel-sgx

[10] Max Hoffmann et al., "Efficient Zero-Knowledge Arguments in the Discrete Log Setting, Revisited," ACM Digital Library, 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3319535.3354251