American Journal of Innovation in Science and Engineering (AJISE)

ISSN: 2158-7205 (ONLINE)

VOLUME 5 ISSUE 2 (2026)

PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

American Journal of Innovation in Science and Engineering (AJISE)

Volume 5 Issue 2, Year 2026
ISSN: 2158-7205 (Online)
DOI: https://doi.org/10.54536/ajise.v5i1.7220
https://journals.e-palli.com/home/index.php/ajise

Security Challenges in AI and Cloud Infrastructure, Including Software Supply Chains and Model-Deployment Pipelines

Emmanuel Olayinka Afolabi[1*], Goodness Nwabueze[2]

Article Information

Received: January 02, 2026
Accepted: March 08, 2026
Published: June 25, 2026

Keywords

AI Supply-Chain Security, Container Risk, Cloud Infrastructure, Dependency, ML Pipeline Integrity, Model Deployment Security

ABSTRACT

AI-enabled services are increasingly being engineered as cloud-native pipelines that reuse dependencies, deliver predictions via managed endpoints, source data, package models and containers. This widens the attack surface across cloud boundaries. This scoping review examines the security challenges associated with cloud and AI infrastructure, including those relating to model deployment pipelines and software supply chains. Using a PRISMA-Scr-guided process and PCC-framed question, studies published between 2015 and 2025 were identified in IEEE Xplore, Scopus, and the ACM Digital Library. These were then de-duplicated and screened. Nineteen empirical studies were synthesized, covering exposure of inference boundaries, data/model compromise, and propagation through imagies, build systems, dependencies, and accelerators in cloud-native environments. The evidence shows that clean-label poisoning can persist with minimal accuracy loss, that gradients and prediction interfaces can enable model theft and leak sensitive training information, and that build artifacts, registry, and container can carry malicious code or secrets at scale. Evaluated controls include runtime entropy checks and trigger reconstruction, confidential inference and privacy-preserving training, GPU isolation and builds, provenance frameworks, detectors and safe deserialization. This research supports the idea of treating AI security as part of continuous pipeline governance, which means using enforceable gates and telemetry in today's complex multicloud operations.

INTRODUCTION

Problem and Context/Background

AI-enabled services are increasingly being engineered as cloud-native pipelines that train models, package artifacts, ingest data, and provide predictions via managed endpoints. This operational shift means that security must now account for learning behaviour under movement of artifacts and active manipulation across organizational and cloud boundaries, not just application code. In adversarial settings, carefully crafted inputs can cause targeted errors while appearing benign to humans, revealing a discrepancy between standard generalization testing and deployment resilience (Papernot *et al.,* 2016). More recent robustness evaluations suggest that prominent defences, including adversarial and robust training, may not generalize to unseen attack families and can underestimate certain perturbation regimes. This implies that security claims may deteriorate as attackers evolve (Garaev *et al.,* 2024). These dynamics justify treating AI security as an ongoing problem intersecting with engineering choices regarding build automation, cloud operations and data sourcing, rather than a one-time model hardening step.

At the same time, the software ecosystem that supports AI development has become a key source of compromise. Supply-chain attacks exploit automated installation and dependency managers' processes to insert malicious code into packages, which are then reused on a large scale. Ohm *et al.* (2020) examined 174 malicious open-source packages distributed via npm, PyPI and RubyGems, demonstrating how common reuse practices can be exploited and emphasizing the need for more advanced defence strategies than just vulnerability scanning. In the Python ecosystem, Guo *et al.* (2023) report that malicious packages often exhibit behaviours such as command execution and information theft and command execution. They also observe that distribution artifacts can persist across mirror servers even after discovery, which complicates the remediation process. A more subtle dependency is introduced when packages embed hardcoded URLs that may later be repointed or hijacked, creating an external control plane for package behaviour (Wyss *et al.,* 2023). Containerization can amplify this risk by freezing dependencies into images and distributing them widely. Longitudinal assessment of Docker images shows that vulnerability counts can increase over time despite regular updates, and tool-to-tool discrepancies can complicate risk interpretation (Mills *et al.,* 2023).

The expansion of cloud infrastructure further increases exposure, as AI workloads increasingly depend on managed services and shared accelerators. Demonstrations of cross-VM covert and side-channel attacks on commercial cloud FPGA instances show that contention over shared related resources and PCIe can leak information and undermine assumptions about tenant isolation, thereby broadening the attack surface beyond standard virtualization abstractions (Giechaskiel *et al.,* 2022). Meanwhile, organizations are adopting MLOps to industrialize model delivery in domains that

[1] Haaga-Helia University of Applied Sciences, Helsinki, Finland
[2] Senior Data scientist, micron technology, Folsom CA, United States
[*] Corresponding author's e-mail: emmanuelolayinkaa@gmail.com

blend physical and digital. Research into MLOps-enabled security strategies in operational technologies reveals that securing ML deployments necessitates the integration of governance, incident handling, and monitoring into the operational lifecycle rather than treating security as an add-on after deployment (Ahmad *et al.,* 2024). Finally, artifacts in AI pipelines behave like executable components. The phenomenon of supply-chain poisoning exemplifies the vulnerability of pre-trained models to being subverted for the purpose of survival in the context of downstream adaptation, thereby elevating model hubs to a position of paramount importance within critical infrastructure (Wang *et al.,* 2025). The application of registry-scale detection has been demonstrated to demonstrate the potential for cross-ecosystem learning to unveil malevolent packages (Zhang *et al.,* 2025). Supply-chain assurances can be invalidated by malicious models, according to industry commentary (Sood & Zeadally, 2025).

Scope of Review

This scoping review maps the empirical evidence on the security challenges spanning AI/ML pipelines and the cloud infrastructure that hosts them. It treats managed inference endpoints, models, dependencies, datasets, and containers as coupled attack surfaces. The review focuses on the operational pipelines involved in serving, fine-tuning and training, including registries, model hubs, CI/CD, and shared accelerators, across hybrid deployments, multi-cloud, and single-cloud, multi-cloud. Using a PCC framework, it synthesizes peer-reviewed studies evaluating compromise pathways and controls, such as backdoors and poisoning, unsafe artifacts and deserialization, container and dependency risks, model theft and privacy leakage, and isolation hardening. The review prioritizes measurable security outcomes and actionable pipeline hardening.

Aim and Specific Objectives of Review

Aim

The aim of this study is for peer-reviewed empirical evidence on compromise pathways,security threats, and evaluated controls across AI/ML supply chains and cloud infrastructure to be mapped and classified. This evidence is to be examined across a range of areas, including data, models, infrastructure isolation, endpoints, dependency/CI-CD stages. The approach to be used is a PCC-framed, PRISMA-ScR-reported scoping review.

Objectives

i. Characterize the compromise of integrity across data and model layers by identifying empirical studies on dataset poisoning, Trojan horses/model backdoors, serialization/ unsafe artifacts, and the controls used to detect or mitigate these risks during onboarding and training.

ii. Examine empirical studies on model extraction/model theft and privacy leakage (e.g. model inversion/membership) in MLaaS/API serving to synthesise evidence on inference-boundary exposure, and appraise protective mechanisms such as confidential inference and privacy-preserving computation and confidential inference.

iii. Evaluate the effectiveness of infrastructure hardening controls and cloud-native supply chain by analysing empirical evidence of dependency chain compromise, image propagation/container, multi-talent acclerator isolation and build/CI-CD provenance. Summarise the operational implications of securing multicloud AI delivery.

MATERIALS AND METHODS

Design and Reporting Standards

This study adopted a scoping review design based on the Population–Concept–Context (PCC) framework, mapping the characteristics, scope, and operational security implications of controls and threats across cloud infrastructure, AI and pipelines. The PRISMA extension for Scoping Reviews (PRISMA-ScR) will be used for reporting, including a transparent study-selection flow diagram and explicit documentation of screening decisions. No protocol registration was undertaken. In line with the objective of scoping reviews, which is to map and classify evidence rather than estimate pooled effects, no formal risk-of-bias assessment was conducted. Methodological safeguards included predefined eligibility criteria, a multi-database strategy, dual-stage screening and standardized data charting.

Eligibility Criteria

Inclusive PCC criteria were applied to identify empirical studies that address security challenges in AI supply chains (including deployment and dependencies pipelines, models and data) and the cloud infrastructure hosting them. Non-empirical and out-of-scope reports were excluded. Peer-reviewed primary studies (e.g. system/defence evaluations, attack demonstrations, applied case studies and measurement studies) were eligible if they explicitly examined compromise pathways or security controls relevant to AI/ML pipelines, cloud multi-tenancy or software supply chains. Table 1 summarises the eligibility criteria.

Table 1: Eligibility Criteria according to the PCC Framework

Item	Inclusion Criteria	Exclusion Criteria
Population	Operational AI/ML systems and pipelines (training, fine-tuning, serving), and the supporting cloud or hybrid infrastructure (e.g., CI/CD, containers, GPUs/FPGAs, registries, model hubs).	Non-AI systems; purely offline software without AI pipeline relevance; purely theoretical constructs not linked to operational pipelines.

Concept	Security threats and controls spanning AI and software supply chains: dataset poisoning, model backdoors/trojans, unsafe model artifacts/serialization, dependency-chain compromise, container/image risks, model theft/extraction, privacy leakage, and infrastructure isolation hardening.	General AI ethics/fairness papers without security or supply-chain mechanisms; generic cybersecurity not tied to AI/cloud pipeline stages; performance-only ML papers.
Context	Any sector and geography; single-cloud, multicloud, and hybrid deployments; MLaaS/API-based serving; open-source ecosystems (e.g., package registries) and model repositories.	Non-cloud contexts where the threat/control cannot be generalized to AI/cloud pipelines (e.g., purely embedded/air-gapped systems without supply-chain components).
Study Designs	Peer-reviewed empirical studies: experimental attack/defence evaluations, systems security papers, measurement studies, implementation studies, and applied case studies.	Narrative reviews, editorials, letters, tutorials, protocols without results, theses, preprints, standards/guidelines documents.
Outcomes	Security-relevant outcomes such as attack success rate/impact, detection performance, false-positive/false-negative behaviour, mitigation effectiveness, overhead/latency, scalability, propagation risk, or operational deployment evidence.	Outcomes unrelated to security properties of AI/cloud pipelines (e.g., accuracy improvements without threat model or control evaluation).
Publication Type	Peer-reviewed journal articles	Preprints, theses, reports, guidelines, books/chapters, grey literature
Language	English	Non-English
Timeframe	2010-2025	Studies published outside this timeframe

Information Sources

The retrieval process was limited to literature that had been reviewed by other experts and listed in three databases chosen to provide comprehensive coverage of research into software engineering, computer security, and AI systems: Scopus, IEEE Xplore, and the ACM Digital Library. No grey literature sources were consulted. Records were exported to Zotero for de-duplication, followed by dual, independent screening at the title/abstract and full-text stages, with any discrepancies resolved through consensus.

Search Strategy

Database-specific strategies were employed, combining free-text terms and database fields representing the PCC elements (AI/ML systems and pipelines, as well as supply-chain and cloud infrastructure security), with sensitivity being prioritized. Filters were applied to restrict the search to English-language, peer-reviewed sources from 2015–2025. Search strings were refined and piloted in order to capture studies spanning inference endpoints, containerization, dataset curation, model artifacts, dependency chains, and infrastructure isolation.

Table 2: Search String

Database	Search string
IEEE Xplore	(("machine learning" OR "deep learning" OR AI OR "pretrained model" OR "model hub") AND ("supply chain" OR dependency OR "package manager" OR PyPI OR npm OR container OR Docker OR "model artifact" OR "model signing" OR "deserialization") AND (poison* OR backdoor OR trojan OR "model stealing" OR extraction OR "membership inference" OR "model inversion" OR "side-channel" OR "multi-tenant")) AND (2015–2025)
ACM Digital Library	(("machine learning" OR "deep learning" OR AI OR "foundation model" OR "pre-trained model") AND ("software supply chain" OR dependency OR registry OR "package ecosystem" OR container OR Docker OR CI/CD OR MLOps OR "model deployment") AND (poisoning OR backdoor OR trojan OR "malicious package" OR typosquat* OR "package confusion" OR "model theft" OR "privacy leakage" OR "trusted execution environment" OR enclave)) AND (2015–2025)

Scopus	TITLE-ABS-KEY (("machine learning" OR "deep learning" OR AI OR "pretrained model" OR "model hub") AND ("supply chain" OR dependency OR registry OR "package manager" OR PyPI OR npm OR container OR Docker OR CI/CD OR MLOps OR "model deployment") AND (poison* OR backdoor OR trojan OR "malicious package" OR typosquat* OR "package confusion" OR "model stealing" OR extraction OR "membership inference" OR "model inversion" OR "side channel" OR multi-tenant OR SGX OR enclave)) AND (LIMIT-TO (LANGUAGE, "English")) AND (PUBYEAR > 2014 AND PUBYEAR < 2026)

Table 2 summarises the search strings used.

Data Extraction and Synthesis

Data charting used a standardized template that was piloted on five to ten studies before being refined. The extracted variables were aligned to the synthesis needs of the review and its three results themes, capturing: (i) bibliographic details, (ii) study type, (iii) the pipeline layer targeted (model artefact, endpoint, data, dependency/ CD/CI or infrastructure), (iv) the security/threat problem, (v) the assets at risk, (vi) the empirical setting (scale, dataset and system, dataset), (vii) the method/ control evaluated (measurment or defence, attack) and (viii) the key findings and practical implications for AI and cloud hardening.

The synthesis was mapping-oriented and descriptive rather than effect-estimating. It included counts by pipeline layer, control/attack class and publication year. Narrative synthesis grouped evidence into the review's three themes: (1) data/model integrity compromise, (2) endpoint exposure and confidentiality/privacy risks, and (3) software/model supply chain and infrastructure hardening. This highlights how threats propagate across interconnected AI and cloud delivery stages.

RESULTS AND DISCUSSIONS

Screening and Selection

The database search yielded 473 records. After de-duplication, 258 duplicate records were removed and 216 records were retained for title–abstract screening. Following screening against the eligibility criteria, 93 articles were advanced to full-text assessment, while 123 records were excluded due to wrong titles and abstracts. At full-text stage, 74 articles were excluded (review papers/ editorials/letters/protocols without results/conference abstracts without full text/preprints/grey literature = 32; non-AI systems or purely offline/non-operational pipeline studies and security-irrelevant outcomes = 17; non-cloud contexts not generalisable to AI/cloud pipelines = 15; AI ethics/fairness or generic cybersecurity not tied to AI/cloud pipeline stages = 10). Finally, 19 studies met all criteria and were synthesized (Fig. 1).

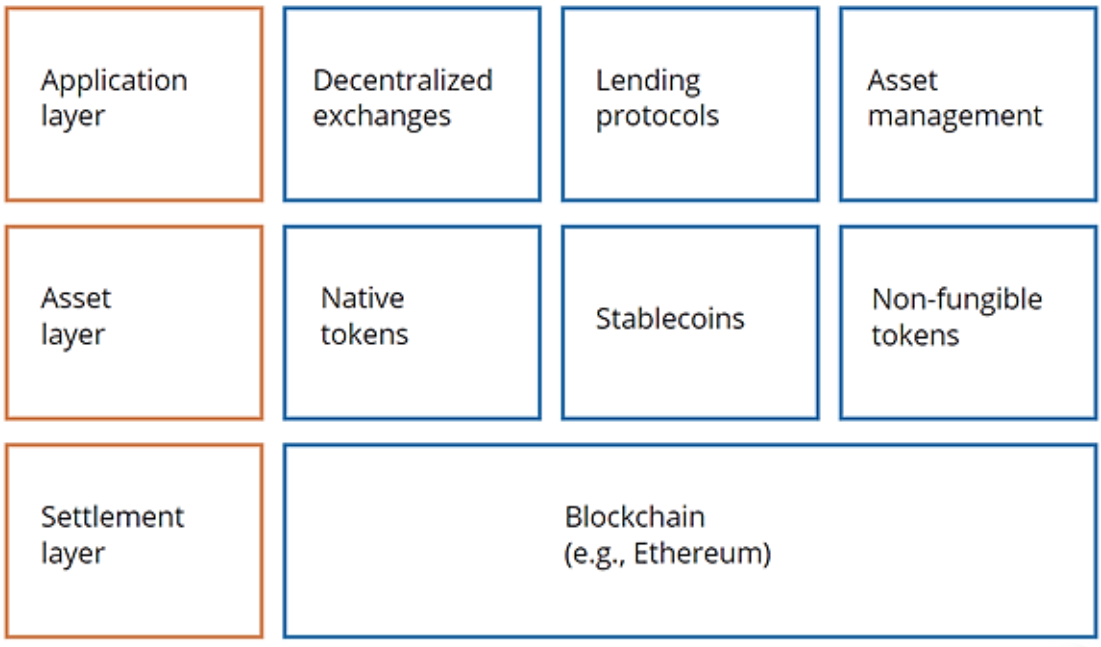

Figure 1: PRISMA-ScR flow diagram of study selection

Summary of Study Characteristics

The evidence base comprised 19 peer-reviewed studies from the fields of security, software engineering and AI systems, published between 2016 and 2025. The designs clustered into three types: (i) pre-trained weights, training signals or inference outputs, experimental attack demonstrations targeting training data; (ii) privacy-preserving computation or mitigation, defensive evaluations proposing detection; and (iii) large-scale ecosystem or infrastructure measurements. Threat surfaces spanned registries and model artifacts, container/build artifacts, dataset curation, dependency chains in package ecosystems, and serving endpoints and cloud accelerators. The empirical settings ranged from standard benchmarks and real machine learning (ML) stacks (e.g. enclave-assisted inference, popular frameworks, common vision datasets) to internet-scale analyses of multi-tenant accelerator studies, container images and package registries. Outcomes were reported as reconstruction fidelity or leakage, false alarms, detector accuracy, attack success rates, operational overhead (latency/throughput) and evidence of real-world reporting or deployment. The studies mapped onto three synthesis themes: integrity compromise, endpoint confidentiality/privacy risk, and software/model supply chain plus infrastructure hardening in production contexts.

Result Synthesis

Theme 1. Data- and Model-Level Compromise: Backdoors, Trojans, and Training-Signal Leakage

Upstream compromise of the AI supply chain is a frequent occurrence. Small manipulations can be embedded into learned behaviour and remain dormant until a trigger appears. The process of 'Trojaning' demonstrates that an attacker can take a shared model, inject malicious behaviour and derive a general trigger by retraining it in a matter of minutes to hours, without requiring access to the original training dataset. This allows them to maintain normal test accuracy and achieve nearly 100% trigger activation (Liu *et al.,* 2018). Clean-label backdoor attacks ensure that poisoned inputs remain consistent with their labels by using adversarial examples and GAN-generated data. This weakens the protection offered by basic data filtering (Turner *et al.,* 2018). As backdoors can remain hidden, model acceptance must include targeted mitigation and detection. Neural Cleanse reconstructs candidate triggers, identifies backdoors, and mitigates them through input filters, neuron pruning and unlearning. It has been demonstrated to be robust against attack variants (Wang *et al.,* 2019). In addition to pre-deployment screening, STRIP performs runtime detection by checking prediction entropy and perturbing incoming inputs. It reported a false acceptance rate of under 1% at a false rejection rate of 1%, and achieved a false rejection rate (FRR) of 0% and a false acceptance rate (FAR) of 0% on GTSRB and CIFAR10 in its evaluations (Gao *et al.,* 2019). Beyond poisoning, distributed and collaborative training can leak raw training data. Deep leakage from gradients can reconstruct original examples from shared gradients in a few steps for both vision and natural language processing (NLP) tasks (Zhu *et al.,* 2019). These findings together justify the implementation of AI-specific controls, such as gatekeeping and model signing, safeguards for gradient sharing, data provenance, and backdoor-focused red-teaming, rather than relying solely on software-based supply-chain checks.

Theme 2. Cloud Inference and MLaaS Exposure: Confidentiality, Membership Inference, and Protected Serving

The included studies demonstrate that deployment shifts key risks to the inference boundary, where attackers can exploit legitimate-looking queries and cloud administrators may be untrusted. Model theft via prediction APIs highlights that publicly accessible prediction interfaces can facilitate the appropriation of a model's functionality in ML-as-a-service settings (Tramèr *et al.,* 2016). Even when the model is not fully extracted, a related privacy failure can occur: membership inference attacks use black-box access to determine whether a record was in the training dataset. These attacks have been evaluated against commercial MLaaS providers (e.g. Amazon and Google), as well as against a hospital discharge dataset (Shokri *et al.,* 2017). Rather than merely adding perimeter controls, two of the included systems respond by changing the computational substrate. SecureML provides privacy-preserving training for neural and regression networks under stochastic gradient descent, using a non-colluding two-server model and secure two-party computation. It reports protocols that are orders of magnitude faster than previous baselines, and which can be scaled to millions of samples with thousands of features (Mohassel & Zhang, 2017). When it comes to serving, Occlumency uses SGX to protect the integrity and confidentiality of user data by enclosing remote deep-learning inference. It has been shown to improve speed by 3.6 times compared to the baseline SGX inference, while maintaining a latency overhead of 72% relative to native inference (Lee *et al.,* 2019). The evidence suggests that cloud AI security is an architectural decision. To reduce confidentiality and privacy leakage, it is necessary to redesign the inference and training pathways. This is because managing operational constraints introduced by MPC and TEEs is also required.

Theme 3. Cloud-Native Supply Chains for AI: Cependencies, Containers, Builds, GPUs, and Model Artifacts

Across the included studies, AI pipelines amplify supply-chain risk because models, packages and images are reused by different teams and in different cloud environments. In-TOTO operationalises end-to-end provenance and provides case studies on 30 supply-chain compromises (Torres-Arias *et al.,* 2019). Package confusion is mapped into 13 mechanisms from over 1,200 documented attacks, and selected detectors achieve high-quality matches with

minimal false positives (Neupane *et al.,* 2023). Cross-ecosystem behaviour-sequence modelling identified 683 new malicious packages in PyPI and 799 in npm (Zhang *et al.,* 2025). The issue of vulnerability debt serves only to exacerbate the situation: a total of 1,396 reports pertaining to vulnerabilities affected 698 PyPI packages, and took more than three years to come to light. By way of comparison, dependent projects can take months to update (Alfadel *et al.,* 2023). Analysis of over 100 TensorFlow vulnerability instances in deep-learning stacks reveals that many issues are rooted in missing or incorrect checks, and the effectiveness of tested static analysis tools is low (Filus & Domańska, 2023). Container images create an additional propagation channel. A study of 337,171 internet-wide images found secrets in 8.5% of them and documented the use of leaked keys in the wild (Dahlmanns *et al.,* 2023). A Docker ecosystem study identified 93.7% of influential images with known malicious images and vulnerabilities with downstream impact (Shi *et al.,* 2025). Monitoring is complemented by hardening: source-to-binary binding is achieved through attestable builds using TEEs (Hugenroth *et al.,* 2025), pickle-based models are safely loaded by PickleBall (Kellas *et al.,* 2025), and multi-tenant risks are limited by GPU bounds checking for real deployments (Pavlidakis *et al.,* 2024).

Table 3: Data Extraction Table for Included Studies

Included study	Liu *et al.,* 2018	Turner *et al.,* 2019	Wang *et al.,* 2019	Gao *et al.,* 2019	Zhu *et al.,* 2019
Primary pipeline layer	Model artefact adoption (pretrained)	Training data	Model validation	Inference monitoring	Distributed training
Threat/ issue	Trojan/backdoor injected post-training via trigger	Clean-label backdoor poisoning	Hidden triggers/ backdoors in DNNs	Trojaned inputs activating backdoors	Training data leakage via shared gradients
Study contribution	Attack method	Attack method	Detection + mitigation	Runtime defense	Attack method
Empirical setting	Trigger reverse-engineering + brief retraining; no original dataset	Adversarial examples + GAN-generated data	Trigger reconstruction; input filters; pruning; unlearning	Entropy test via intentional perturbations	Reconstruct training data from gradients
Key quantitative evidence	Triggered ~100% with normal accuracy; minutes–hours	Poisoned samples appear label-consistent (stealth)	Efficacy shown across DNNs; robust to variants	FAR <1% at FRR 1%; 0% FRR/FAR on CIFAR10 & GTSRB	Recovers original data in few steps (CV & NLP)
Real-world relevance	Highlights safety-critical consequences; motivates model vetting	Shows filtering-by-label is insufficient	Practical "model acceptance gate" approach	Supports continuous monitoring of deployed models	Reframes gradients as sensitive artifacts
Control implication	Pre-deployment backdoor tests; model provenance/ signing	Data provenance + robust curation; red-teaming for clean-label	Backdoor scanning + mitigation before deployment	Add behavioral monitoring + incident response playbooks	Secure aggregation / privacy mechanisms; limit gradient exposure

Zhang *et al.*, 2025	Neupane *et al.*, 2023	Torres-Arias *et al.*, 2019	Pavlidakis *et al.*, 2024	Lee *et al.*, 2019	Mohassel & Zhang, 2017	Shokri *et al.*, 2017	Tramèr *et al.*, 2016
Dependencies	Dependencies	Build/CI-CD	Infrastructure (GPU)	Inference (TEE)	Training (privacy)	Inference API	Inference API
Malicious OSS packages in npm/ PyPI	Package confusion (beyond typosquatting)	Software supply-chain tampering across steps	Cross-tenant GPU memory-safety risks	Data leakage by attackers/admins during cloud inference	Confidential training across data owners	Membership inference (record presence in training set)	Model extraction/theft via prediction APIs
Detector model	Empirical + detectors	Framework/ defense	System/defense	System/defense	System/defense	Attack + evaluation	Attack study
Behavior-sequence features; fine-tuned PLM	13 confusion mechanisms; detectors; npm-scale evaluation	Cryptographic chain-of-custody for pipeline steps	PTX-level bounds checking; partitioning; intercept calls; kernel instrumentation	SGX enclave inference; memory-aware optimizations (Caffe)	2-server non-colluding 2PC; SGD; MPC-friendly nonlinears	Train inference model from black-box outputs	Black-box querying of deployed models
Detected 683 (PyPI) & 799 (npm) new malicious pkgs; 707 acknowledgements	Survey: 77% matches confusing; ~1 warning per 100M+ pairs	Demonstrated on 30 compromises; used in products/ projects	4–12% overhead (avg 9%) on Caffe/ PyTorch workloads	3.6× faster than baseline SGX; 72% latency overhead vs native	Orders-of-magnitude faster than prior; scales to millions, thousands features	Vulnerable models incl. Google/ Amazon MLaaS; mitigation tested	Feasibility demonstrated
Evidence for cross-ecosystem detection transfer	Improves detection of confusable names in the wild	Shows provenance works across cloud-native apps	Practical multi-tenant isolation for accelerators	Demonstrates confidentiality+integrity for remote inference	Evidence that privacy-preserving training can scale	Concrete privacy risk for MLaaS users	Shows IP and security models are exposed by APIs
Deploy registry-level detectors and response workflows	Third-party risk controls for registries and CI	Adopt pipeline provenance and verification policies	GPU isolation controls in multi-tenant clouds	Consider TEEs for sensitive inference workloads	Use MPC where data sharing is constrained	Privacy risk assessment + mitigation in deployment	API governance + leakage-aware serving

Kellas *et al.*, 2025	Hugenroth *et al.*, 2025	Shi *et al.*, 2025	Dahlmanns *et al.*, 2023	Filus & Domańska, 2022	Alfadel *et al.*, 2023
Model artefact loading	Build pipeline	Container ecosystem	Container images	ML framework dependencies	Dependencies
RCE/malware via pickle-based model files	Opaque builds breaking source-to-binary trust	Vulns, secrets, misconfigs, malicious images and propagation	Secrets leakage (keys, API tokens) in images	TensorFlow vulnerability patterns	Vulnerable packages and slow patch uptake (PyPI)
System/defense	System/defense	Ecosystem measurement	Internet-wide measurement	Vuln analysis	Empirical measurement
Static policy generation + dynamic enforcement (drop-in pickle)	TEEs + sandboxed containers; formal model	DITector; 12M repos; dependency graph; 33,952 influential images	337,171 images + 8,076 registries; key reuse analysis	100+ instances; CWE + ODC; tool evaluation	1,396 vuln reports; 2,224 GitHub projects; lifespan analysis
Loads 79.8% benign; rejects 100% malicious; addresses 44.9% pickle prevalence	42s startup; 14% build duration increase; instant verification	93.7% with known vulns; 4,437 secret leaks; 24 malicious; 334 downstream affected	8.5% images contain secrets; 52,107 keys; 275,269 hosts reuse leaked keys	Static analysis low effectiveness; many missing/ incorrect checks	>3 years to discover; 40.86% fixed after disclosure; ~7 months update
Directly targets model-hub ingestion risk	Practical step toward trustworthy build artifacts	Demonstrates propagation and remediation engagement	Direct link between supply chain and active compromise	Signals framework-level risk to ML apps	Shows exposure windows are long
Safe deserialization policies + scanning before promotion	Adopt attestable builds for critical components	Continuous image scanning + dependency graph monitoring	Secret scanning + build hygiene + key rotation	Secure coding + targeted testing for ML libraries	Dependency governance + SBOM/VEX + patch SLAs

Discussion

Summary of Key Findings

Across the 19 studies, evidence converged on the idea of a layered risk picture. Integrity attacks exploit weak controls over model artifacts and training data, while confidentiality attacks share training signals and exploit exposed endpoints. Backdoors can remain dormant during normal testing, preserving accuracy while potentially causing malfunction, so benchmark validation alone is insufficient for safe onboarding. Even in well-generalized models, privacy leakage persists, and updates or gradients can enable membership inference or record reconstruction, thus reinforcing the need for privacy-by-design in distributed learning. The impact of software supply chains is amplified through containers, registries, and builds; vulnerable packages, malicious dependencies, and embedded secrets propagate into AI stacks. Controls were effective when implemented as pipeline safeguards, such as secure loading and artifact scanning, verifiable builds and cryptographic attestation, behavioural monitoring for unusual API usage, and hardware- or cryptography-based protections for sensitive inference or training processes. Reported trade-offs include overhead, attacker adaptation and partial coverage, leaving gaps in areas such as standardized provenance,secure, feedback-driven retraining and cross-cloud enforcement.

Comparison with Global Literature

Early insertion of compromise into the data-model layer is possible, with the potential for this to persist into deployment. The occurrence of trigger-conditioned misbehaviour with minimal accuracy loss has been demonstrated in the cases of trojaning and clean-label poisoning (Liu *et al.,* 2018; Turner *et al.,* 2019). International

operations increase the danger with several entry points or multiple activation mechanisms, and with assaults on third-party language models incorporated in middleware (Xue *et al.,* 2020; Dong *et al.,* 2023). Consequently, the incorporated defences in the included studies (entropy-based screening and trigger reconstruction/mitigation), with the wider shift towards behavioural testing (Wang *et al.,* 2019; Gao *et al.,* 2019). Privacy leakage also occurs upstream: gradient-based reconstruction (Zhu *et al.,* 2019) supports the idea that models can memorize and reveal training content (Song *et al.,* 2017), including that from collaborative updates and white-box access (Melis *et al.,* 2019; Nasr *et al.,* 2019).

At the inference boundary, our findings confirm that MLaaS APIs present a surface through which IP can be lost and privacy harmed in the supply chain. Model extraction remains feasible via prediction interfaces (Tramèr *et al.,* 2016), and membership inference can be successful against deployed models (Shokri *et al.,* 2017). Exposing confidence scores amplifies what attackers can recover: model inversion can infer sensitive attributes and reconstruct recognizable faces, while suppressing or coarsening confidence scores can reduce leakage with limited loss of utility (Fredrikson *et al.,* 2015). Detection evidence is convergent: PRADA detects model theft by analyzing query distribution patterns (Juuti *et al.,* 2019), and Copycat CNN demonstrates that random images can facilitate copying (Correia-Silva *et al.,* 2021). The defensive recommendations from the included studies emphasize layered controls. Privacy-preserving training or aggregation can reduce leakage (Abadi *et al.,* 2016; Bonawitz *et al.,* 2017), while confidential inference and secure computation can provide additional protection where necessary (Mohassel & Zhang, 2017; Lee *et al.,* 2019).

Finally, the supply chain theme in the included studies demonstrates how risk is propagated through dependencies, containers, builds, GPUs and model artifacts in cloud-native reuse. Provenance tooling enables verifiable handovers (Torres-Arias *et al.,* 2019) and aligns with package manager measurements that identify and report malicious packages (Duan *et al.,* 2021). The typosquatting and confusion defences documented in the wider literature are consistent with our findings within the ecosystem (Taylor *et al.,* 2020; Neupane *et al.,* 2023; Zhang *et al.,* 2025). The set of container measurements from the results shows widespread secrets and vulnerable images with downstream impact (Dahlmanns *et al.,* 2023; Shi *et al.,* 2025). Advances in hardening, such as attestable builds, safe deserialization, and GPU isolation, support enforceable gates for artifacts (Hugenroth *et al.,* 2025; Kellas *et al.,* 2025; Pavlidakis *et al.,* 2024). Model-hub evidence operationalizes the 'models as code' risk (Zhao *et al.,* 2024) in multicloud environments.

Implications for Policy and Practice

Policy

It is proposed that a commissioned standard for AI supply-chain security should be established within AI, cloud and software procurement. This would move beyond conventional SBOM-only compliance to include pipeline assurance and AI artifacts. This would entail mandatory provenance requirements for training code, containers, datasets, and model artifacts, as well as verification and cryptographic signing at registry and load time. Furthermore, there would be contractual third-party controls for dependency ecosystems, model hubs, and MLaaS, accompanied by explicit shared-responsibility clauses for multicloud delivery. Sectoral and regulatory governance should require routine red-teaming of AI systems prior to release and at major model updates, as well as auditability baselines and minimum logging across malicious packages, Incident and tenant disclosure triggers for poisoned artifacts, suspicious model behaviour or secret leakage. Quality indicators should be defined, including the percentage of artifacts with verifiable provenance, attestation/ signing coverage across builds, the time taken to revoke compromised artifacts, and the mean time to detect pipeline anomalies. In settings with limited resources, policy should prioritize enforceable baselines, such as default-deny dependency policies, trusted registries, and managed security controls, supported by lightweight audits and standardized templates.

Practice

It is essential to implement a secure 'ingest–build–verify–deploy–monitor' MLOps model, with gated controls applied at every stage from data intake to inference. Data pipelines should enforce lineage capture, integrity checks, source allowlists, and quarantine workflows for anomalous inputs. Model onboarding should require signed artifacts, reproducible builds, safe deserialization, and registry enforcement. Dependency chains should be scanned, pinned, and continuously evaluated for secrets, malicious behaviour, and vulnerable transitive packages. Deployment should be hardened via isolated runtimes (including accelerator isolation where applicable), least-privilege IAM, rate- limited and strict secret management, behaviour drift and monitored inference endpoints with automated alerting for query patterns resembling extraction. Rather than making ad hoc manual model swaps and using unsanctioned dependencies, operational teams should use canary deployments, controlled release channels, post-deployment continuous evaluation, and rollback playbooks. Outcomes should be monitored using reliability KPIs and security, such as anomaly detection precision, model integrity verification failures, attestation pass rate, false-alarm burden, privacy-test results and time to contain suspected compromise. These findings should be fed into iterative pipeline improvements and targeted workforce training.

Limitations of Review

This scoping review is limited by its reliance on peer-reviewed literature, which tends to under-represent the operational incidents that are often reported in

post-mortems and advisories. The included evidence is heterogeneous, featuring different benchmarks, threat models, and metrics. This prevents quantitative synthesis and limits comparability. Several studies evaluate controls in constrained laboratory settings, with limited longitudinal validation in production multicloud environments. The screening process was restricted to selected databases and English publications, so relevant regional or industry-specific work may have been overlooked.

CONCLUSION

This scoping review synthesised 19 empirical studies examining security challenges in AI and cloud infrastructure supply chains. These chains span model artefacts, container and dependency ecosystems, training data, model artefacts, build/CI pipelines, inference endpoints and shared accelerators. The evidence from diverse operational settings shows that integrity compromises, such as trojaned model artefacts and clean-label poisoning, can persist with minimal accuracy loss. It also shows that training signals and inference can enable model theft and privacy leakage, and that cloud-native reuse can propagate risk through registries, images, packages, and opaque builds. However, most of the evaluated controls were tested under limited assumptions and with little longitudinal validation across multicloud environments, and there was incomplete coverage of the full artefact lifecycle. Overall, the evidence suggests that the security of AI/ML is strengthened when safe loading, artefact verification, provenance, isolation and behavioural monitoring are implemented as continuous pipeline gates rather than ad hoc checks. Future work should prioritise standardised provenance for verifiable builds, data and models, and signing with rapid revocation, and containment playbooks across tenants, and field validation of detection. There should also be a focus on interoperable governance metrics, including attestation coverage, time to detect anomalies, time to revoke compromised artefacts and false alarm burden, in order to develop AI security as a dependable capability for cloud-scale deployment.

REFERENCES

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

Sood, A. K., & Zeadally, S. (2025). Malicious AI models undermine software supply-chain security. *Communications of the ACM, 68*(6), 62–71. https://doi.org/10.1145/3704724

Ahmad, T., Adnan, M., Rafi, S., Akbar, M. A., & Anwar, A. (2024, June). MLOps-Enabled Security Strategies for Next-Generation Operational Technologies. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering* (pp. 662-667). https://doi.org/10.1145/3661167.3661283

Alfadel, M., Costa, D.E. & Shihab, E. Empirical analysis of security vulnerabilities in Python packages. *Empir Software Eng 28, 59*(2023). https://doi.org/10.1007/s10664-022-10278-4

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017, October). Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1175-1191).https://doi.org/10.1145/3133956.3133982

Correia-Silva, J. R., Berriel, R. F., Badue, C., De Souza, A. F., & Oliveira-Santos, T. (2021). Copycat CNN: Are random non-labeled data enough to steal knowledge from black-box models?. *Pattern Recognition, 113,* 107830.https://doi.org/10.1016/j.patcog.2021.107830

Dahlmanns, M., Sander, C., Decker, R., & Wehrle, K. (2023, July). Secrets revealed in container images: an internet-wide study on occurrence and impact. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security* (pp. 797-811).https://doi.org/10.1145/3579856.3590329

Dong, P., Guo, S., & Wang, J. (2023, August). Investigating trojan attacks on pre-trained language model-powered database middleware. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 437-447). https://doi.org/10.1145/3580305.3599395

Duan, R., Alrawi, O., Kasturi, R., Elder, R., Saltaformaggio, B., & Lee, W. (2021). Towards measuring supply chain attacks on package managers for interpreted languages. In *Proceedings of the Network and Distributed System Security Symposium (NDSS) 2021*. Internet Society. https://doi.org/10.14722/ndss.2021.23055

Filus, K., & Domańska, J. (2023). Software vulnerabilities in TensorFlow-based deep learning applications. *Computers & Security, 124,* 102948. https://doi.org/10.1016/j.cose.2022.102948

Fredrikson, M., Jha, S., & Ristenpart, T. (2015, October). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1322-1333). https://doi.org/10.1145/2810103.2813677

Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., & Nepal, S. (2019, December). Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference* (pp. 113-125). https://doi.org/10.1145/3359789.3359790

Garaev, R., Rasheed, B., & Khan, A. M. (2024). Not so robust after all: Evaluating the robustness of deep neural networks to unseen adversarial attacks. *Algorithms, 17*(4), 162. 10.3390/a17040162

Giechaskiel, I., Tian, S., & Szefer, J. (2022). Cross-VM covert-and side-channel attacks in cloud FPGAs.

ACM Transactions on Reconfigurable Technology and Systems, 16(1), 1-29. https://doi.org/10.1145/3534972

Guo, W., Xu, Z., Liu, C., Huang, C., Fang, Y., & Liu, Y. (2023, September). An empirical study of malicious code in pypi ecosystem. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 166-177). IEEE. https://doi.org/10.1109/ASE56229.2023.00135

Hugenroth, D., Lins, M., Mayrhofer, R., & Beresford, A. R. (2025, November). Attestable builds: compiling verifiable binaries on untrusted systems using trusted execution environments. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (pp. 4514-4528). https://doi.org/10.1145/3719027.3765128

Juuti, M., Szyller, S., Marchal, S., & Asokan, N. (2019, June). PRADA: protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 512-527). IEEE. doi: 10.1109/EuroSP.2019.00044.

Kellas, A. D., Christou, N., Jiang, W., Li, P., Simon, L., David, Y., ... & Yang, J. (2025, November). PickleBall: Secure Deserialization of Pickle-based Machine Learning Models. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (pp. 3341-3355). https://doi.org/10.1145/3719027.3765037

Lee, T., Lin, Z., Pushp, S., Li, C., Liu, Y., Lee, Y., Xu, F., Xu, C., Zhang, L., & Song, J. (2019). Occlumency: Privacy-preserving remote deep-learning inference using SGX. In *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19)* (pp. 1–15). Association for Computing Machinery. https://doi.org/10.1145/3300061.3345447

Liu, Y., Ma, S., Aafer, Y., Lee, W. C., Zhai, J., Wang, W., & Zhang, X. (2018, January). Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc. https://doi.org/10.14722/ndss.2018.23291.

Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 691–706). IEEE. https://doi.org/10.1109/SP.2019.00029

Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 19–38). IEEE. https://doi.org/10.1109/SP.2017.12

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. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 739–753). IEEE. https://doi.org/10.1109/SP.2019.00065

Neupane, S., Holmes, G., Wyss, E., Davidson, D., & De Carli, L. (2023). Beyond typosquatting: an in-depth look at package confusion. In *32nd USENIX Security Symposium (USENIX Security 23)* (pp. 3439-3456).

Mills, A., White, J., & Legg, P. (2023). Longitudinal risk-based security assessment of docker software container images. *Computers & Security, 135,* 103478. https://doi.org/10.1016/j.cose.2023.103478

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019, January). Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 220-229). https://doi.org/10.1145/3287560.3287596

Ohm, M., Plate, H., Sykosch, A., & Meier, M. (2020). Backstabber's knife collection: A review of open source software supply chain attacks. In C. Maurice, L. Bilge, G. Stringhini, & N. Neves (Eds.), *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2020)* (Lecture Notes in Computer Science, Vol. 12223). Springer. https://doi.org/10.1007/978-3-030-52683-2_2

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 372–387). IEEE. https://doi.org/10.1109/EuroSP.2016.36

Pavlidakis, M., Vasiliadis, G., Mavridis, S., Argyros, A., Chazapis, A., & Bilas, A. (2024, December). Guardian: Safe GPU sharing in multi-tenant environments. In *Proceedings of the 25th International Middleware Conference* (pp. 313–326).

Shi, H., Ying, L., Chen, L., Duan, H., Liu, M., & Xue, Z. (2025, April). Dr. Docker: A Large-Scale Security Measurement of Docker Image Ecosystem. In *Proceedings of the ACM on Web Conference 2025* (pp. 2813-2823). https://doi.org/10.1145/3696410.3714653

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017, May). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)* (pp. 3-18). IEEE. https://doi.org/10.1109/sp.2017.41

Song, C., Ristenpart, T., & Shmatikov, V. (2017, October). Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security* (pp. 587-601). https://doi.org/10.1145/3133956.3134077

Taylor, M., Vaidya, R., Davidson, D., De Carli, L., & Rastogi, V. (2020, November). Defending against package typosquatting. In *International conference on network and system security* (pp. 112-131). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-65745-1_7

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM, 64*(12), 86–92. https://doi.org/10.1145/3458723

Torres-Arias, S., Afzali, H., Kuppusamy, T. K., Curtmola, R., & Cappos, J. (2019). in-toto: Providing farm-to-table guarantees for bits and bytes. In *28th USENIX Security Symposium (USENIX Security 19)* (pp. 1393-1410). https://dl.acm.org/doi/10.5555/3361338.3361435

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)* (pp. 601-618).

Turner, A., Tsipras, D., & Madry, A. (2018). *Clean-label backdoor attacks*.

Uchida, Y., Nagai, Y., Sakazawa, S., & Satoh, S. I. (2017, June). Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval* (pp. 269-277).https://doi.org/10.1145/3078971.3078974

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., & Zhao, B. Y. (2019, May). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)* (pp. 707-723). IEEE.

Wang, H., Guo, S., He, J., Liu, H., Zhang, T., & Xiang, T. (2025, April). Model Supply Chain Poisoning: Backdooring Pre-trained Models via Embedding Indistinguishability. In *Proceedings of the ACM on Web Conference 2025* (pp. 840-851). https://doi.org/10.1145/3696410.3714624

Wyss, E., Davidson, D., & De Carli, L. (2023, November). What's in a URL? An Analysis of Hardcoded URLs in npm Packages. In *Proceedings of the 2024 Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses* (pp. 26-32). https://doi.org/10.1145/3689944.3696168

Xue, M., He, C., Wang, J., & Liu, W. (2020). One-to-n & n-to-one: Two advanced backdoor attacks against deep learning models. *IEEE Transactions on Dependable and Secure Computing, 19*(3), 1562-1578. 10.1109/TDSC.2020.3028448.

Zhang, J., Huang, K., Huang, Y., Chen, B., Wang, R., Wang, C., & Peng, X. (2025). Killing two birds with one stone: Malicious package detection in NPM and PyPI using a single model of malicious behavior sequence. *ACM Transactions on Software Engineering and Methodology, 34*(4), 1-28.

Zhao, J., Wang, S., Zhao, Y., Hou, X., Wang, K., Gao, P., ... & Wang, H. (2024, October). Models are codes: Towards measuring malicious code poisoning attacks on pre-trained model hubs. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering* (pp. 2087-2098). https://doi.org/10.1145/3691620.3695271

Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems, 32*.

Zhang, J., Huang, K., Huang, Y., Chen, B., Wang, R., Wang, C., & Peng, X. (2025). Killing two birds with one stone: Malicious package detection in NPM and PyPI using a single model of malicious behavior sequence. *ACM Transactions on Software Engineering and Methodology, 34*(4), 1-28. https://doi.org/10.1145/3705304