databricks

Whitepaper

Databricks AI Security Framework (DASF)

Version 2.0

Table of Contents

Executive Summary

Machine learning (ML) and generative AI (GenAI) are transforming the future of work by enhancing innovation, competitiveness and employee productivity. However, organizations are grappling with the dual challenge of leveraging AI technologies while managing potential security and privacy risks, such as data leakage breaches, increased liability due to misuse and regulatory noncompliance.

Adopting AI raises regulatory considerations, exemplified by evolving U.S. federal government mandates, the EU AI Act and over 120 additional countries with AI-related laws, underscoring the importance of responsible governance and oversight. The evolving legal and regulatory landscape, combined with uncertainties around ownership accountability, leaves data, IT, business, governance and security leaders navigating how to effectively harness generative AI for organizational benefits. They must also balance the potential negative impacts of those risks with positive organizational outcomes.

Databricks created the **Databricks AI Security Framework (DASF)** to help organizations address the evolving risks associated with the global adoption of AI. Unlike approaches focusing solely on securing models or endpoints, the DASF adopts a comprehensive strategy to mitigate risks in AI systems. Based on real-world evidence indicating that attackers employ simple tactics to compromise ML-driven systems, the DASF offers actionable defensive control recommendations. These recommendations can be updated as new risks emerge and additional controls become available. The framework's development involved a thorough review of multiple risk management frameworks, recommendations, whitepapers, policies and AI security acts.

The DASF is designed to foster collaboration between business, IT, data, AI, governance and security teams throughout the AI lifecycle. It addresses the evolving nature of data science from a research-oriented discipline to a project-based discipline and facilitates structured conversations on security threats and mitigations without needing deep expertise crossover. This document is targeted to security teams, ML practitioners, business leaders and governance officers, providing insights into how ML impacts system security, applying security engineering principles to ML and offering a detailed guide for understanding the security risks, potentially negative impacts, mitigation controls and compliance of specific ML systems.

The DASF walks readers through 12 foundational components of a generic data-centric AI system (Figure 1): raw data, data prep, datasets, data and AI governance, machine learning algorithms, evaluation, machine learning models, model management, model serving and inference, inference response, machine learning operations, and data and AI platform security. Databricks identified 62 technical security risks that arise across these components and dedicated a chapter describing the specific component, the associated risks, risk impacts and the available controls we recommend you leverage. We also provide a guide to each AI and ML mitigation control — its shared responsibility between Databricks and your organization, and the associated Databricks technical documentation available to learn how to enable said control. We've created a compendium document for the DASF, now available for download. This compendium is a versatile resource for practitioners to engage with DASF by organizing and applying its risks, threats, controls and mappings to industry-recognized standards, including MITRE, OWASP, NIST, ISO, HITRUST and more. The intent behind releasing the compendium is to ease operationalization of the DASF.

The framework concludes with recommendations from Databricks on how to manage and deploy AI models safely and securely, which are consistent with the core tenets of machine learning adoption: identify the ML business use case, determine the ML deployment model, select the most pertinent risks, enumerate threats for each risk and choose which controls to implement. We also provide further reading to enhance your knowledge of the AI field and the frameworks we reviewed as part of our analysis.

While we strive for accuracy, given the evolving nature of AI, please feel free to contact us with any feedback or suggestions. Your input is valuable to us. If you want to participate in one of our AI Risk workshops, please contact dasf@databricks.com. If you're curious about how Databricks approaches security, please visit our Security and Trust Center.

What's new in the Databricks AI Security Framework 2.0?

We formed an AI workgroup by partnering with experts and thought leaders in the industry and worked extensively on further studying security risks and controls for AI system components. As a result, in Databricks AI Security Framework 2.0, we introduced compound AI systems into AI system components and added seven new risks and five new controls. We mapped AI risks and controls to 10 industry standards and frameworks. We documented control mapping to shared responsibility, novelty and more metadata to help organizations easily understand and apply controls. We updated the framework to include worksheets to help determine appropriate mitigation controls based on use case, AI deployment model, risks, risk impacts, business impacts, standards and compliance requirements to help organizations prioritize controls based on their risk appetite as they deploy AI applications in their enterprises.

Introduction

Machine learning (ML) and generative AI (GenAI) are revolutionizing the future of work. Organizations understand that AI is helping to build innovation, maintain competitiveness and improve the productivity of their employees. Equally, organizations understand that their data provides a competitive advantage for their AI applications. This naturally leads to an intense focus on models as the core component of AI applications and speculation on the capabilities of future models. However, as more developers build on top of large language models (LLMs), they're realizing that compound AI systems are required to obtain state-of-the-art AI results. Leveraging these technologies presents opportunities but also potential risks. There's a risk of security and privacy breaches, as the data sent to an external LLM could be leaked or summarized. Several organizations have even banned the use of LLMs due to sensitive enterprise data being sent by users. Organizations are also concerned about potential hazards such as data loss, breach of data confidentiality, model theft and risks of ensuring existing and evolving compliance and regulation when they use their data for ML and GenAI. Without the proper access controls, users can use generative AI models to find confidential data they shouldn't have access to. If the models are customer-facing, one organization might accidentally receive data related to a different organization. Or a skilled attacker can extract data they shouldn't have access to. Without the auditability and traceability of these models and their data, organizations face compliance risks.

AI adoption also brings a crucial regulatory dimension, emphasizing the need for thoughtful oversight and responsible governance. The National Institute of Standards and Technology (NIST) recently published their Artificial Intelligence Risk Management Framework (AI RMF) to help federal agencies manage and secure their information systems. It provides a structured process for identifying, assessing and mitigating cybersecurity risks. Gartner's 2023 Security Leader's Guide to Data Security report[1] predicts that "at least one global company will see its AI deployment banned by a regulator for noncompliance with data protection or AI governance legislation by 2027." With ownership accountability and an ever-evolving legal and regulatory landscape, data, IT and security leaders are still unclear on how to take advantage of generative AI for their organization while mitigating any perceived risks.

[1] Gartner, Security Leader's Guide to Data Security, Andrew Bales. September 7, 2023.

Databricks developed the **Databricks AI Security Framework (DASF)** to help organizations understand how AI can be safely realized and risks mitigated as the global community incorporates AI into more systems. The DASF takes a holistic approach to awareness and mitigation of AI security risks instead of focusing only on the security of models or model endpoints. Abundant real-world evidence suggests that attackers use simple tactics to subvert ML-driven systems. That is why, with the DASF, we propose actionable defensive control recommendations. These recommendations are subject to change as new risks are identified and new controls are made available. We reviewed many risk management frameworks, recommendations, whitepapers, policies and acts on AI security. We encourage the audience to review such material, including some of the material linked in the resources section of this document. Your feedback is welcome.

1.1 Intended audience

The Databricks AI Security Framework is intended to be used by data and AI teams collaborating with their security teams across the AI/ML lifecycle. Traditionally, the skill sets of data scientists, data engineers, security teams, governance officers and DevSecOps (development, security and operations) engineering teams didn't overlap. The communication gap between data scientists and these teams was manageable, given the research-oriented nature of data science and its primary focus on delivering information to executives. However, as data science transforms into a project-based discipline, it becomes crucial for these teams to collaborate.

The guidance in this document provides a way for disciplines to have structured conversations on these new threats and mitigations without requiring security engineers to become data scientists or vice versa. We mostly did this work for our customers to ensure the security and compliance of production ML use cases on the Databricks Data Intelligence Platform. That said, we believe that what we've produced will be helpful to four major audience groups:

1. **Security teams (CISOs, security leaders, DevSecOPs, site reliability engineers [SREs])** can use the DASF to understand how ML will impact the security of systems they may be asked to secure, as well as to understand some of the basic mechanisms of ML.
2. **ML practitioners and engineers (data engineers, data architects, ML engineers, data scientists)** can use the DASF to understand how security engineering and, more specifically, the "secure by design" mentality can be applied to ML.

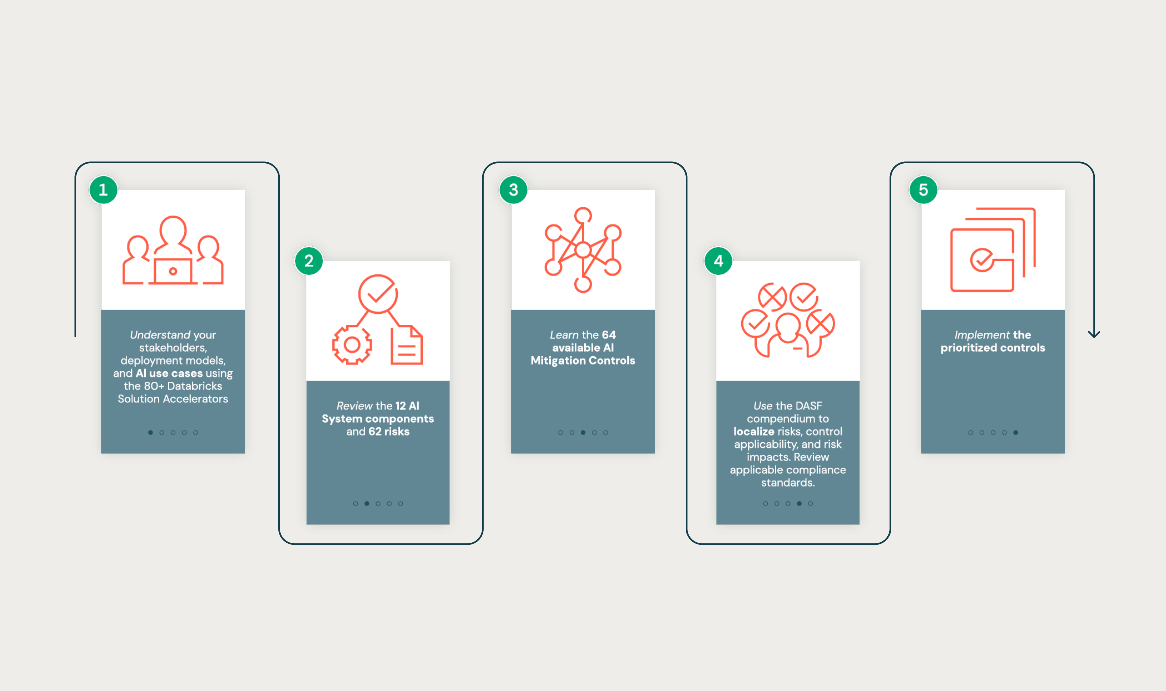
Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

3. **Governance leaders, risk officers, legal teams and policymakers** can use the DASF as a detailed guide into a risk mindset to learn more about the security and compliance of specific ML systems.
4. **Business leaders (CIOS, CTOs, CDOs, Board of Directors)** can use the DASF as a guide to gain awareness of how AI security risk has a direct link to business risk and the associated negative impacts.

If you're new to GenAI, you can build foundational knowledge, including large language models (LLMs), with four short videos in this Generative AI Fundamentals course created by Databricks. In this free training, you'll learn what generative AI is, what the main generative AI applications are and their capabilities and potential applications across various domains. It will also cover the limits and risks of generative AI technologies, including ethical considerations.

1.2 How to use this document

The Databricks AI Security Framework (DASF) provides a path to understanding, evaluating and mitigating risks, ensuring confident deployment of AI use cases. This section provides a high-level overview of the framework, while the paper dives deeper into the steps outlined in the figure below. In the Conclusion, we'll demonstrate how to apply the DASF to your AI use cases. We also built a companion video to this whitepaper, Introducing the Databricks AI Security Framework (DASF) to Manage AI Security Risks, to make it easy to understand the DASF and reduce the friction getting started with it.


1
Understand your stakeholders, deployment models, and **AI use cases** using the 80+ Databricks Solution Accelerators
2
Review the **12 AI System components** and **62 risks**
3
Learn the **64 available AI Mitigation Controls**
4
Use the DASF compendium to **localize** risks, control applicability, and risk impacts. Review applicable compliance standards.
5
Implement **the prioritized controls**

To get started, we suggest that organizations find out what type of AI deployment models are being used. As a guideline, we define deployment model broadly as the following:

- **Predictive ML models:** These are traditional structured data machine learning models trained on your enterprise tabular data. They're typically Python models packaged in the MLflow format. These ML models can be trained using standard ML libraries like scikit-learn, XGBoost, PyTorch and Hugging Face transformers and can include any Python code.
- **Frontier models** made available by Foundation Model APIs: These models are curated foundation model architectures that support optimized inference. Base models like Meta Llama, GTE-Large and Mixtral are available for immediate use with pay-per-token pricing, and workloads that require performance guarantees and fine-tuned model variants can be deployed with provisioned throughput. We subcategorize the usage patterns of these models as Foundation Model APIs to LLMs and retrieval augmented generation (RAG), pretraining and fine-tuning use of LLMs.
- **External models (third-party services):** These are models that are hosted outside of Databricks. Endpoints that serve external models can be centrally governed, and customers can establish rate limits and access control for them. Examples include foundation models such as OpenAI's GPT, Anthropic's Claude and others.

Next, we recommend that organizations identify where in their organization AI systems are being built, the process and who's responsible. The modern AI system lifecycle often involves diverse stakeholders, including business stakeholders, subject matter experts, governance officers, data engineers, data scientists, research scientists, application developers, administrators, AI security engineers, DevSecOps engineers and MLSecOps engineers. The Databricks AI Security Framework is designed for collaborative use throughout the AI lifecycle, fostering closer collaboration between data and AI teams and their security counterparts to enhance the overall security of AI systems.

We recommend that those responsible for AI systems begin by reviewing the 12 foundational components of a generic data-centric AI system and the types of AI models, as outlined in Section 2: Risks in AI System Components. This section details security risk considerations, AI risk impacts, business impacts, applicable AI deployment models and potential mitigation controls for each component, helping organizations reduce overall risk in their AI system development and deployment processes. Each security risk is mapped to a set of mitigation controls that are ranked in prioritized order, starting with the perimeter security to data security. These guidelines apply to providers of all AI systems, whether built from scratch or using third-party tools and services, and encompass both predictive ML models and generative AI models.

Once the relevant risks are identified, teams can determine which controls are applicable by model type from the comprehensive list in Section 3: Understanding AI Risk Mitigation Controls. Each control is tagged as “Out-of-the-box,” “Configuration” or “Implementation,” helping teams estimate the effort involved in the implementation of the control, with reference links to relevant documentation provided. The controls in this document are applicable for all teams, even if they don’t use Databricks to build their use cases. That said, we’ll refer to documentation or features in Databricks terminology where it allows us to simplify our language or make this document more actionable for our direct customers. We hope those who don’t use Databricks will also be able to follow along without issue.

Finally, to further support and guide those responsible for AI systems and to help organizations achieve strategic and business goals, we’ve suggested potential risk impacts for each of the 12 components and 62 technical security risks. Risk impacts have been grouped into two categories: initial AI and business impacts, closely aligning our guidance to the FAIR risk methodology of primary and secondary losses. You can leverage the compendium document for the Databricks AI Security Framework (DASF), now available for download (Google sheet, Excel). This compendium is a versatile resource for practitioners to engage with DASF by organizing and applying its risks, threats, controls and mappings to industry-recognized standards, including MITRE, OWASP, NIST, ISO, HITRUST and more. The intent behind releasing the compendium is to ease operationalization of the DASF.

Our experience shows that implementing these guidelines helps customers build AI systems that are more secure and functional.

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

Risks in AI System Components

The DASF starts with a generic AI system in terms of its constituent components and works through generic system risks. By understanding the components, how they work together and the risk analysis of such architecture, an organization concerned about security can get a jump start on determining risks in its specific AI system. The Databricks Security team considered these risks and built mitigation controls into our Databricks Data Intelligence Platform. We mapped the respective Databricks Platform control and link to Databricks product documentation for each risk.

AI system components

12 canonical components of an end-to-end AI system

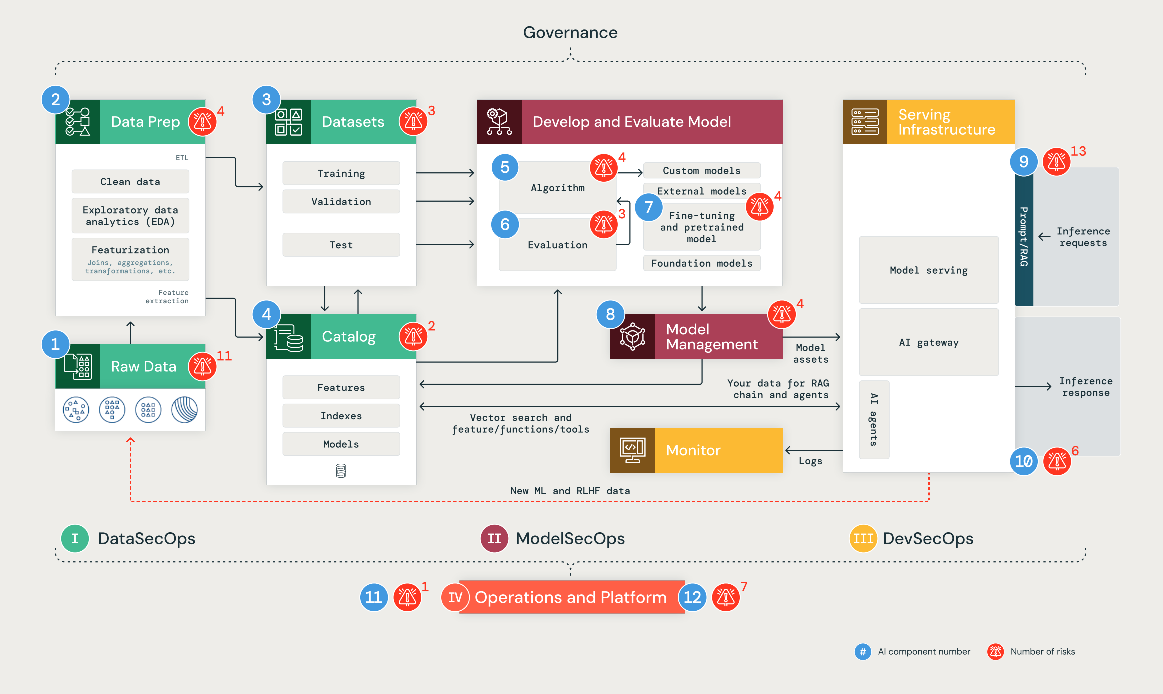


Figure 1: Foundational components of a generic data-centric AI system

Figure 1 shows the 12 foundational components of a generic data-centric AI and ML system, broadly categorized into four major stages:

I. **Data operations (#1–#4 in Figure 1)** include ingesting and transforming data and ensuring data security and governance. Good ML models depend on reliable data pipelines and secure DataOps infrastructure.

II. **Model operations (#5–#8 in Figure 1)** include building predictive ML models, acquiring models from a model marketplace or using LLMs like OpenAI or Foundation Model APIs. Developing a model requires a series of experiments and a way to track and compare the conditions and results of those experiments.

III. **Model deployment and serving (#9 and #10 in Figure 1)** consists of securely building model images, isolating and securely serving models, automated scaling, rate limiting and monitoring deployed models. Additionally, it includes feature and function serving, a high-availability, low-latency service for structured data in retrieval augmented generation (RAG) applications, as well as features that are required for other applications, such as models served outside of the platform or any other application that requires features based on data in the catalog.

IV. **Operations and platform (#11 and #12 in Figure 1)** include platform vulnerability management and patching, model isolation and controls to the system and authorized access to models with security in the architecture. Also included is operational tooling for CI/CD. It ensures the complete lifecycle meets the required standards by keeping the distinct execution environments — development, staging and production — for secure MLOps.

Compound AI systems

Compound AI systems, as defined by the Berkeley AI Research (BAIR) blog, leverage the strengths of multiple AI models, tools and pipelines to enhance performance, versatility and reusability compared to solely using individual models. AI-forward organizations are building compound AI systems to deliver complex solutions. For example, RAG applications with agent frameworks are compound AI systems, combining at least one model with contextual enrichment (see Figure 1) rooted in your enterprise data. A compound AI system requires advanced functionality like context switching, state management and multi-step flow management capabilities, which increases the attack surface. The additional controls needed to secure these systems are addressed in this version.

In our analysis of AI systems, we identified 62 technical security risks across the 12 AI components based on the AI model types deployed by our customers (namely, predictive ML models, generative foundation models and external models as described above), customer questions and questionnaires, security reviews of customer deployments, in-person AI risk workshops and customer surveys about AI risks. In the table below, we outline these basic components that align with steps in any AI system and highlight the types of security risks our team identified.

AI SYSTEM STAGE	AI SYSTEM COMPONENTS (FIGURE 1)	POTENTIAL SECURITY RISKS
Data operations	1 Raw data → 2 Data preparation → 3 Datasets → 4 Catalog and governance →	20 specific risks: 1.1 Insufficient access controls → 1.2 Missing data classification → 1.3 Poor data quality → 1.4 Ineffective storage and encryption → 1.5 Lack of data versioning → 1.6 Insufficient data lineage → 1.7 Lack of data trustworthiness → 1.8 Legality of data → 1.9 Stale data → 1.10 Lack of data access logs → 1.11 Compromised third-party datasets → 2.1 Preprocessing integrity → 2.2 Feature manipulation → 2.3 Raw data criteria → 2.4 Adversarial partitions → 3.1 Data poisoning → 3.2 Ineffective storage and encryption → 3.3 Label flipping → 4.1 Lack of traceability and transparency of model assets → 4.2 Lack of end-to-end ML lifecycle →
Model operations	5 ML algorithm → 6 Evaluation → 7 Model build → 8 Model management →	15 specific risks: 5.1 Lack of tracking and reproducibility of experiments → 5.2 Model drift → 5.3 Hyperparameters stealing → 5.4 Malicious libraries → 6.1 Evaluation data poisoning → 6.2 Insufficient evaluation data → 6.3 Lack of interpretability and explainability → 7.1 Backdoor machine learning/Trojaned model → 7.2 Model assets leak → 7.3 ML supply chain vulnerabilities → 7.4 Source code control attack → 8.1 Model attribution → 8.2 Model theft → 8.3 Model lifecycle without HITL → 8.4 Model inversion →

SYSTEM STAGE	SYSTEM COMPONENTS (FIGURE 1)	POTENTIAL SECURITY RISKS
Model deployment and serving	9 Model Serving — inference requests → 10 Model Serving — inference responses →	19 specific risks: 9.1 Prompt inject → 9.2 Model inversion → 9.3 Model breakout → 9.4 Looped input → 9.5 Infer training data membership → 9.6 Discover ML model ontology → 9.7 Denial of service (DOS) → 9.8 LLM hallucinations → 9.9 Input resource control → 9.10 Accidental exposure of unauthorized data to models → 9.11 Model Inference API access → 9.12 LLM jailbreak → 9.13 Excessive agency → 10.1 Lack of audit and monitoring inference quality → 10.2 Output manipulation → 10.3 Discover ML model ontology → 10.4 Discover ML model family → 10.5 Black box attacks → 10.6 Sensitive data output from a model →
Operations and platform	11 ML operations → 12 ML platform →	8 specific risks: 11.1 Lack of MLOps — repeatable enforced standards → 12.1 Lack of vulnerability management → 12.2 Lack of penetration testing, red teaming and bug bounty → 12.3 Lack of incident response → 12.4 Unauthorized privileged access → 12.5 Poor SDLC → 12.6 Lack of compliance → 12.7 Initial access →

The 12 foundational components of a generic data-centric AI/ML model, security risks and risk impact considerations are discussed in detail below.

Note: We're aware of nascent risks such as energy-latency attacks, rowhammer attacks, side-channel attacks, evasion attacks, functional adversarial attacks and other adversarial examples, but these are out of scope for this version of the framework. We may reconsider these and any new novel risks in later versions if we see them becoming material.

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

2.1 Raw data

Data is the most important aspect of AI systems because it provides the foundation that all AI functionality is built on. Raw data includes enterprise data, metadata and operational data. It can be semi-structured or unstructured such as images, sensor data or documents. This data can be batch data or streaming data. Data security is paramount and equally important for ensuring the security of machine learning algorithms and any technical deployment particulars. Securing raw data is a challenge in its own right, and all data collections in an AI system are subject to the usual data security challenges and some new ones. A fully trained machine learning (ML) system, whether online or offline, will inevitably encounter new input data during normal operations or retraining processes. Fine-tuning and pretraining of LLMs further increases these risks by allowing customizations with potentially sensitive data.

RISK/DESCRIPTION	MITIGATION CONTROLS
RAW DATA 1.1 **Insufficient access controls** Effective access management is fundamental to data security, ensuring only authorized individuals or groups can access specific datasets. Such security protocols encompass authentication, authorization and finely tuned access controls tailored to the scope of access required by each user, down to the file or record level. Establishing definitive governance policies for data access is imperative in response to the heightened risks from data breaches and regulations like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). These policies guard against unauthorized use and are a cornerstone of preserving data integrity and maintaining customer trust. Data operations →	DASF 1 SSO with IdP and MFA to authenticate and limit who can access your data and AI platform DASF 2 Sync users and groups to inherit your organizational roles to authorize access to data DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources DASF 51 Share data and AI assets securely DASF 55 Monitor audit logs to collaborate in a secure environment DASF 59 Use clean rooms to collaborate in a secure environment **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○
RAW DATA 1.2 **Missing data classification** Data classification is critical for data governance, enabling organizations to effectively sort and categorize data by sensitivity, importance and criticality. As data volumes grow exponentially, prioritizing sensitive information protection, risk reduction and data quality becomes imperative. Classification facilitates the implementation of appropriate security measures and governance policies by evaluating data's risk and value. A robust classification strategy strengthens data governance, mitigates risks and ensures data integrity and security on a scalable level. Data operations →	DASF 6 Classify data with tags as it's ingested into the platform aligning with the organization's governance requirements **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION | MITIGATION CONTROLS

RAW DATA 1.3

Poor data quality

Data quality is crucial for reliable data-driven decisions and is a cornerstone of data governance. Malicious actors threaten data integrity, accuracy and consistency, challenging the analytics and decision-making processes that depend on high-quality data, just as a well-intentioned user with poor-quality data can limit the efficacy of an AI system. To safeguard against these threats, organizations must rigorously evaluate key data attributes — accuracy, completeness, freshness and rule compliance. Prioritizing data quality enables organizations to trace data lineage, apply data quality rules and monitor changes, ensuring analytical accuracy and cost-effectiveness.

Data operations →

DASF 7 Enforce data quality checks on batch and streaming datasets

DASF 21 Monitor data and AI system from a single pane of glass

DASF 36 Set up monitoring alerts

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RAW DATA 1.4

Ineffective storage and encryption

Insecure data storage leaves organizations vulnerable to unauthorized access, potentially leading to data breaches with significant legal, financial and reputational consequences. Encrypting data at rest can help to render the data unreadable to unauthorized actors who bypass security measures or attempt large-scale data exfiltration. Additionally, compliance with industry-specific data security regulations often necessitates such measures.

Data operations →

DASF 5 Control access to data and other objects for metadata encryption across all data assets

DASF 8 Encrypt data at rest

DASF 9 Encrypt data in transit

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RAW DATA 1.5

Lack of data versioning

When data gets corrupted by a malicious user by introducing a new set of data or by corrupting a data pipeline, you'll need to be able to roll back or trace back to the original data.

Data operations →

DASF 10 Version data and track change logs on large-scale datasets that are fed to your models

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RAW DATA 1.6

Insufficient data lineage

Because data may come from multiple sources and go through multiple transformations over its lifecycle, understanding data transparency and usage requirements in AI training is important to risk management. Many compliance regulations require organizations to have a clear understanding and traceability of data used for AI. Data lineage helps organizations be compliant and audit-ready, thereby alleviating the operational overhead of manually creating the trails of data flows for audit reporting purposes.

Data operations →

DASF 11 Capture and view data lineage

DASF 51 Share data and AI assets securely

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION	MITIGATION CONTROLS
RAW DATA 1.7 **Lack of data trustworthiness** Attackers may tamper with or poison raw input data (training data, RAG data, etc.). Adversaries may exploit public datasets, which often resemble those used by targeted organizations. To mitigate these threats, organizations should validate data sources, implement integrity checks and utilize AI and machine learning for anomaly detection. Data operations →	DASF 10 Version data and track change logs on large-scale datasets that are fed to your models DASF 51 Share data and AI assets securely DASF 59 Use clean rooms to collaborate in a secure environment **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○
RAW DATA 1.8 **Legality of data** Intellectual property concerns of training data and and legal mandates — such as those from GDPR, CCPA and LGPD — necessitate the capability of machine learning systems to "delete" specific data. But you often can't "untrain" a model; during the training process, input data is encoded into the internal representation of the model, characterized by elements like thresholds and weights, which could become subject to legal constraints. Tracking your training data and retraining your model using clean and ownership-verified datasets is essential for meeting regulatory demands. Data operations →	DASF 12 Delete records from datasets and retrain models to forget data subjects DASF 27 Pretrain a large language model (LLM) to only use the data that's allowed with LLMs for inference DASF 29 Build MLOps workflows to track models and trace data sources and lineage to retrain models with the updated dataset by following legal constraints **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○
RAW DATA 1.9 **Stale data** When downstream data isn't timely or accurate, business processes can be delayed, significantly affecting overall efficiency. Attackers may deliberately target these systems with attacks like denial of service, which can undermine the model's performance and dependability. It's crucial to proactively counteract these threats. Data streaming and performance monitoring help protect against such risks, maintaining the input data integrity and ensuring they're delivered promptly to the model. Data operations →	DASF 7 Enforce data quality checks on batch and streaming datasets DASF 13 Use near real-time data for fault-tolerant, near real-time data ingestion, processing and machine learning, and AI for streaming data **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○

RISK/DESCRIPTION	MITIGATION CONTROLS
RAW DATA 1.10 **Lack of data access logs** Without proper audit mechanisms, an organization may not be fully aware of their risk surface area, leaving them vulnerable to data breaches and regulatory noncompliance. Therefore, a well-designed audit team within a data governance or security governance organization is critical in ensuring data security and compliance with regulations such as GDPR and CCPA. By implementing effective data access auditing strategies, organizations can maintain the trust of their customers and protect their data from unauthorized access or misuse. Data operations →	DASF 14 Audit actions performed on datasets DASF 55 Monitor audit logs to track access to data, AI and other resources **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○
RAW DATA 1.11 **Compromised third-party datasets** Adversaries may poison training data and publish it to public locations or manipulate source datasets known to be used in the AI system. The poisoned dataset may be a novel dataset or a poisoned variant of an existing open source dataset. This data may be introduced to a victim system via supply chain compromise. Data operations →	DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources DASF 7 Enforce data quality checks on batch and streaming datasets for data sanity checks and automatically detect anomalies before they make it to the datasets DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources DASF 17 Track and reproduce the training data used for ML model training and identify ML models and runs derived from a particular dataset **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○

2.2 Data prep

Machine learning algorithms require raw input data to be transformed into a representational form they can understand. This data preparation step can impact the security and explainability of an ML system, as data plays a crucial role in security. Data preparation includes the following tasks:

1. **Cleaning and formatting data** includes handling missing values or outliers, ensuring data is in the correct format and removing unneeded columns.
2. **Preprocessing data** includes tasks like numerical transformations, aggregating data, encoding text or image data, and creating new features.
3. **Combining data** includes tasks like joining tables or merging datasets.
4. **Labeling data** includes tasks like identifying raw data (images, text files, videos and so on) and adding one or more meaningful and informative labels to provide context so an ML model can learn from it.
5. **Validating and visualizing data** includes exploratory data analysis to ensure data is correct and ready for ML. Visualizations like histograms, scatter plots, box and whisker plots, line plots and bar charts are all useful tools to confirm data correctness.

RISK/DESCRIPTION	MITIGATION CONTROLS
DATA PREP 2.1 **Preprocessing integrity** Preprocessing includes numerical transformations, data aggregation, text or image data encoding, and new feature creation, followed by combining data by joining tables or merging datasets. Data preparation involves cleaning and formatting tasks such as handling missing values, ensuring correct formats and removing unnecessary columns. Insiders or external actors can introduce errors or manipulate data during preprocessing or from the information repository itself. Data operations →	DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform DASF 2 Sync users and groups to inherit your organizational roles to access data DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources DASF 7 Enforce data quality checks on batch and streaming datasets for data sanity checks and automatically detect anomalies before they make it to the datasets DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources DASF 15 Explore datasets and identify problems DASF 16 Secure model features to reduce the risk of malicious actors manipulating the features that feed into ML training DASF 42 Employ data-centric MLOps and LLMOps to promote models as code DASF 52 Source code control to control and audit your knowledge object integrity DASF 55 Monitor audit logs

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

RISK/DESCRIPTION — MITIGATION CONTROLS

DATA PREP 2.2

Feature manipulation

In almost all cases, raw data requires preprocessing and transformation before it's used to build a model. This process, known as *feature engineering*, involves converting raw data into structured features, the building blocks of the model. Feature engineering is critical to quality and effectiveness of the model. However, how data are annotated into features can introduce the risk of incorporating attacker biases into an AI/ML system. This can compromise the integrity and accuracy of the model and is a significant security concern for models used in critical decision-making (e.g., financial forecasting, fraud detection).

Data operations →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests
- DASF 16 Secure model features to prevent and track unauthorized updates to features and for lineage or traceability
- DASF 42 Employ data-centric MLOps and LLMOps to promote models as code

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

DATA PREP 2.3

Raw data criteria

An attacker who understands raw data selection criteria may be able to introduce malicious input that compromises system integrity or functionality later in the model lifecycle. Exploitation of this knowledge allows the attacker to bypass established security measures and manipulate the system's output or behavior. Implementing stringent security measures to safeguard against such manipulations is essential for maintaining the integrity and reliability of ML systems.

Data operations →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists to restrict the IP addresses that can authenticate to Databricks
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 42 Employ data-centric MLOps and LLMOps for unit and integration testing
- DASF 43 Use access control lists to control access to data, data streams and notebooks

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

RISK/DESCRIPTION

DATA PREP 2.4

Adversarial partitions

If an attacker can influence the partitioning of datasets used in training and evaluation, they can effectively exercise indirect control over the ML system by making them vulnerable to adversarial attacks, where carefully crafted inputs lead to incorrect outputs. These attacks can exploit the space partitioning capabilities of machine learning models, such as tree ensembles and neural networks, leading to misclassifications even in high-confidence scenarios. This form of "model control" can lead to biased or compromised outcomes. Therefore, it's crucial that datasets accurately reflect the intended operational reality of the ML system. Implementing stringent security measures to safeguard against such manipulations is essential for maintaining the integrity and reliability of ML systems.

Data operations →

MITIGATION CONTROLS

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists to restrict the IP addresses that can authenticate to Databricks
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 17 Track and reproduce the training data used for ML model training to track and reproduce the training data partitions and the human owner accountable for ML model training, as well as identify ML models and runs derived from a particular dataset
- DASF 42 Employ data-centric MLOps and LLMOpsfor unit and integration testing

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

2.3 Datasets

Prepared data must be grouped into different datasets: a training set, a validation set and a testing set. The training set is used as input to the machine learning algorithm. The validation set is used to tune hyperparameters and to monitor the machine learning algorithm for overfitting. The test set is used after learning is complete to evaluate performance.

When creating these groupings, special care must be taken to avoid predisposing the ML algorithm to future attacks, such as adversarial partitions. In particular, the training set deeply influences an ML system's future behavior. Manipulating the training data represents a direct and potent means of compromising ML systems. By injecting malicious or adversarial samples into the training set, attackers can subtly influence the model's behavior, potentially leading to misclassification, performance degradation or even security breaches.

These approaches, often called "data poisoning" or "backdoor attacks," pose a significant threat to the robustness and reliability of ML systems deployed in various critical domains. Dataset security concerns with foundation models include the potential for leaks of sensitive information. Fine-tuning and pretraining of LLMs further increases these risks as it allows customizations with sensitive data.

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

DATASETS 3.1

Data poisoning

Attackers can compromise an ML system by contaminating its training data to manipulate its output at the inference stage. All three initial components of a typical ML system — raw data, data preparation and datasets — are susceptible to poisoning attacks. Intentionally manipulated data, possibly coordinated across these components, derail the ML training process and create an unreliable model. Practitioners must assess the potential extent of training data an attacker might control internally and externally and the resultant risks.

Data operations →

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data

DASF 3 Restrict access using IP access lists to restrict the IP addresses that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 7 Enforce data quality checks on batch and streaming datasets for data sanity checks, and automatically detect anomalies before they make it to the datasets

DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources

DASF 14 Audit actions performed on datasets

DASF 16 Secure model features

DASF 17 Track and reproduce the training data used for ML model training and identify ML models and runs derived from a particular dataset

DASF 51 Share data and AI assets securely

DASF 55 Monitor audit logs to track access to data, AI and other resources

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

DATASETS 3.2

Ineffective storage and encryption

Data stored and managed insecurely pose significant risks, especially for ML systems. It's crucial to consider who has access to training datasets and the reasons behind this access. While access controls are a vital mitigation strategy, their effectiveness is limited with public data sources, where traditional security measures may not apply. Therefore, it's essential to ask: What are the implications if an attacker gains access and control over your data sources? Understanding and preparing for this scenario is critical for safeguarding the integrity of ML systems.

Data operations →

DASF 5 Control access to data and other objects for metadata encryption across all data assets

DASF 8 Encrypt data at rest

DASF 9 Encrypt data in transit

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION | MITIGATION CONTROLS

DATASETS 3.3

Label flipping

Label-flipping attacks are a distinctive type of data poisoning where the attacker manipulates the labels of a fraction of the training data. In these attacks, the attacker changes the labels of specific training points, which can mislead the ML model during training. Even with constrained capabilities, these attacks have been shown to significantly degrade the system's performance, demonstrating their potential to compromise the accuracy and reliability of ML models.

Data operations →

DASF 5 Control access to data and other objects for metadata encryption across all data assets

DASF 8 Encrypt data at rest

DASF 9 Encrypt data in transit

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

2.4 Data catalog governance

Data catalog and governance is a comprehensive approach that comprises the principles, practices and tools to manage an organization's data assets throughout their lifecycle. Managing governance for data and AI assets enables centralized access control, auditing, lineage, data and model discovery capabilities, and allows organizations to limit the risk of data or model duplication, improper use of classified data for training, loss of provenance and model theft.

Additionally, if sensitive information in datasets is inadequately secured, breaches and leaks can expose personally identifiable information (PII), financial data and even trade secrets, and cause potential legal repercussions, reputational damage and financial losses.

Proper data catalog governance allows for audit trails and tracing the origin and transformations of data used to train AI models. This transparency encourages trust and accountability, reduces risk of biases and improves AI outcomes.

databricks

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

RISK/DESCRIPTION

MITIGATION CONTROLS

GOVERNANCE 4.1

Lack of traceability and transparency of model assets

The absence of traceability in data, model assets and models and the lack of accountable human oversight pose significant risks in machine learning systems. This lack of traceability can:

- Undermine the supportability and adoption of these systems, as it hampers the ability to maintain and update them effectively
- Impact trust and transparency, which are essential for users to understand and rely on the system's decisions
- Limit the organization's ability to meet regulatory, compliance and legal obligations, as these often require clear documentation and tracking of data and model-related processes

Data operations →

- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources
- DASF 7 Enforce data quality checks on batch and streaming datasets for data sanity checks, and automatically detect anomalies before they make it to the datasets
- DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources
- DASF 16 Secure model features
- DASF 17 Track and reproduce the training data used for ML model training and identify ML models and runs derived from a particular dataset
- DASF 18 Govern model assets for traceability
- DASF 55 Monitor audit logs

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

GOVERNANCE 4.2

Lack of end-to-end ML lifecycle

Continuously measure, track and analyze key metrics, such as performance, accuracy and user engagement, to ensure the AI system's reliability. Demonstrating consistent performance builds trustworthiness among users, customers and regulators.

Data operations →

- DASF 19 Manage end-to-end machine learning lifecycle for measuring, versioning, tracking model artifacts, metrics and results
- DASF 21 Monitor data and AI system from a single pane of glass
- DASF 42 Employ data-centric MLOps and LLMOps unit and integration testing

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

2.5 Machine learning algorithms

The process of running a machine learning algorithm on a dataset (called *training data*) and optimizing the algorithm to find certain patterns or outputs is called *model training*. During training, the machine learning algorithm is optimized to find certain patterns or outputs from the dataset, depending on the task. While the machine learning algorithm forms the technical core of any ML system, attacks against it generally present significantly less security risk compared to the data used for training, testing and eventual operation. However, it's crucial to recognize and mitigate certain security risks associated with the choice of algorithm and its operational mode.

How machine learning algorithms process data primarily falls into two broad categories: offline and online. Offline systems are trained on a fixed dataset, "frozen" and subsequently used for predictions with new data. This approach is particularly common for classification tasks. Conversely, online systems continuously learn and adapt through iterative training with new data.

From a security perspective, offline systems possess certain advantages. Their fixed, static nature reduces the attack surface and minimizes exposure to data-borne vulnerabilities over time. In contrast, online systems are constantly exposed to new data, potentially increasing their susceptibility to poisoning attacks, adversarial inputs and manipulation of learning processes. Therefore, the choice between offline and online learning algorithms should be made carefully, considering the ML system's specific security requirements and operating environment.

RISK/DESCRIPTION	MITIGATION CONTROLS
ALGORITHMS 5.1 **Lack of tracking and reproducibility of experiments** ML development is often poorly documented and tracked, and results that can't be reproduced may lead to overconfidence in an ML system's performance. Common issues include: ▪ Critical details missing from a model's description ▪ Results that are fragile, producing dramatically different results on a different GPU (even one that is supposed to be spec-identical) ▪ Extensive tweaks to the authors' system until it outperforms the untweaked "baseline," resulting in asserted improvements that aren't borne out in practice (particularly common in academic work) Additionally, adversaries may gain initial access to a system by compromising the unique portions of the ML supply chain. This could include the model itself, training data or its annotations, parts of the ML software stack, or even GPU hardware. In some instances, the attacker will need secondary access to fully carry out an attack using compromised supply chain components. Model operations →	DASF 20 Track ML training runs for documenting, measuring, versioning, tracking model artifacts, including algorithms, training environment, hyperparameters, metrics and results DASF 42 Employ data-centric MLOps and LLMOps to promote models as code and automate ML tasks for cross-environment reproducibility DASF 55 Monitor audit logs to track access to data, AI and other resources **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○ Pretrained LLMs: ○ Foundational models: ○ External models: ○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION

MITIGATION CONTROLS

ALGORITHMS 5.2

Model drift

Model drift in machine learning systems can occur due to changes in feature data or target dependencies. This drift can be broadly classified into three scenarios:

- **Concept drift:** where the statistical properties of the target variable change over time
- **Data drift:** involving changes in the distribution of input data
- **Upstream data changes:** occur due to alterations in data collection or processing methods before the data reaches the model

Clever attackers can exploit these scenarios to evade an ML system for adversarial purposes.

Model operations →

DASF 16 Secure model features to track changes to features

DASF 17 Track training data with MLflow and Delta Lake to track upstream data changes

DASF 21 Monitor data and AI system from a single pane of glass for changes and take action when changes occur. Have a feedback loop from a monitoring system and refresh models over time to help avoid model staleness.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

ALGORITHMS 5.3

Hyperparameters stealing

Hyperparameters in machine learning are often deemed confidential due to their commercial value and role in proprietary learning processes. If attackers gain access to these hyperparameters, they may steal or manipulate them — altering, concealing or even adding hyperparameters. Such unauthorized interventions can harm the ML system, compromising performance and reliability or revealing sensitive algorithmic strategies.

Model operations →

DASF 20 Track ML training runs in the model development process, including parameter settings, securely

DASF 42 Employ data-centric MLOps and LLMOps employing separate model lifecycle stages by Unity Catalog schema

DASF 43 Use access control lists via workspace access controls

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

ALGORITHMS 5.4

Malicious libraries

Attackers can upload malicious libraries to public repositories that have the potential to compromise systems, data and models. Administrators should manage and restrict the installation and usage of third-party libraries, safeguarding systems, pipelines and data. This risk may also manifest in 2.2 Data prep in exploratory data analysis (EDA).

Model operations →

DASF 53 Third-party library control to limit the potential for malicious third-party libraries and code to be used on mission-critical workloads

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	●	Foundational models:	○	External models:	○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

2.6 Evaluation

Assessing the effectiveness of a machine learning system in achieving its intended functionalities is a critical step in its development cycle. Post-learning evaluation utilizes dedicated datasets to systematically analyze the performance of a trained model on its specific task.

RISK/DESCRIPTION | MITIGATION CONTROLS

EVALUATION 6.1

Evaluation data poisoning

Upstream attacks against data, where the data is tampered with before it's used for machine learning, significantly complicate the training and evaluation of ML models. Poisoning of the evaluation data impacts the model validation and testing process. These attacks can corrupt or alter the data in a way that skews the training process, leading to unreliable models.

Model operations →

DASF 1	SSO with IdP and MFA to limit who can access your data and AI platform
DASF 2	Sync users and groups to inherit your organizational roles to access data
DASF 3	Restrict access using IP access lists to restrict the IP addresses that can authenticate to your data and AI platform
DASF 4	Restrict access using private link as strong controls that limit the source for inbound requests
DASF 5	Control access to data and other objects for permissions model across all data assets to protect data and sources
DASF 7	Enforce data quality checks on batch and streaming datasets for data sanity checks, and automatically detect anomalies before they make it to the datasets
DASF 11	Capture and view data lineage to capture the lineage all the way to the original raw data sources
DASF 42	Employ data-centric MLOps and LLMOps unit and integration testing
DASF 44	Trigger actions in response to a specific event via automated jobs to notify human-in-the-loop (HITL)
DASF 45	Evaluate models to capture performance insights for language models
DASF 49	Automate LLM evaluation

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

EVALUATION 6.2

Insufficient evaluation data

Evaluation datasets can also be too small or too similar to the training data to be useful. Poor evaluation data can lead to biases, hallucinations and toxic output. It's difficult to effectively evaluate large language models (LLMs), as these models rarely have an objective ground truth labeled. Consequently, organizations frequently struggle to determine the trustworthiness of these models in critical, unsupervised use cases, given the uncertainties in their evaluation.

Model operations →

DASF 22	Build models with all representative, accurate and relevant data sources to evaluate on clean and sufficient data
DASF 25	Use retrieval augmented generation (RAG) with large language models (LLMs)
DASF 45	Evaluate models to capture performance insights for language models
DASF 47	Compare LLM outputs on set prompts to assess LLM project with an interactive prompt interface

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	●

RISK/DESCRIPTION

MITIGATION CONTROLS

EVALUATION 6.3

Lack of interpretability and explainability

A lack of interpretability and explainability poses a unique challenge due to the scale and complexity of frontier or GenAI LLMs. The degree to which we can understand the internal workings of a model and trace its decision-making process refers to interpretability. The ability to provide a human-understandable explanation of a model's outputs or decision-making process refers to explainability. Both interpretability and explainability are required to develop trustworthiness in AI systems. They're also required to assist with understanding and mitigating security risks such as prompt injection, model inversion, misuse, data breach, data poisoning, adversarial attacks, security auditing and regulatory compliance.

References:
Mechanistic Interpretability for AI Safety — A Review

Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks

Model operations →

DASF 35 Track model performance to evaluate quality

DASF 37 Set up inference tables for monitoring and debugging prompts

DASF 42 Employ data-centric MLOps and LLMOps unit and integration testing

DASF 45 Evaluate models to capture performance insights for language models

Additional controls for consideration:

- Continuous benchmark testing of features throughout the model lifecycle in prediction accuracy, traceability and decision understanding. LIME, SHAP, DEEPLIFT and dashboards of factors influencing decisions.
- Isolate functionality using a modular architecture
- Implement explainable AI tools and techniques that can generate human-understandable explanations for model outputs
- Model cards or a data sheet that provides information on data training sets, known biases, evaluation tests, intended uses and unintended uses. From the perspective of how the AI developer trained the model or how your organization has modified the model if your organization has fine-tuned that model.
- Red teaming and adversarial machine learning
- Human-in-the-loop or human-at-the-helm

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	●

2.7 Machine learning models

A machine learning model is a program that can find patterns or make decisions from a previously unseen dataset. During training, the machine learning algorithm is optimized to find certain patterns or outputs from the dataset, depending on the task. The output of this process — often a computer program with specific rules and data structures — is called a machine learning model.

Deploying a fully trained machine learning model to production introduces several critical risks to address. Notably, some risks discussed in the previous section on evaluation risks, such as overfitting, directly apply here. Open source or commercial models, not trained within your organization, carry the same risks with the added challenge that your organization lacks control over the model's development and training.

Additionally, external models may be Trojan horse back doors or harboring other uncontrolled risks, depriving you of the competitive advantage of leveraging your own data and potentially exposing your data to unauthorized access. Therefore, it's crucial to carefully consider and mitigate these potential risks before deploying any pretrained model to production.

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL 7.1 **Backdoor machine learning/ Trojaned model** There are inherent risks when using public ML/LLM models or outsourcing their training, akin to the dangers associated with executable (.exe) files. A malicious third party handling the training process could tamper with the data or deliver a "Trojan model" that intentionally misclassifies specific inputs. Additionally, open source models may contain hidden malicious code that can exfiltrate sensitive data upon deployment. These risks are pertinent in both external models and outsourced model development scenarios, necessitating scrutiny and verification of models before use. Model operations →	DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform DASF 5 Control access to data and other objects DASF 19 Manage end-to-end machine learning lifecycle DASF 23 Register, version, approve, promote and deploy models and scan models for malicious code when using third-party models or libraries DASF 34 Run models in multiple layers of isolation. Models are considered untrusted code: Deploy models and custom LLMs with multiple layers of isolation. DASF 42 Employ data-centric MLOps and LLMOps to promote models as code using CI/CD. Scan third-party models continuously to identify hidden cybersecurity risks and threats such as malware, vulnerabilities and integrity issues to detect possible signs of malicious activity, including malware, tampering and back doors. DASF 43 Use access control lists to limit who can bring models and limit the use of public models DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses **Additional controls to consider:** Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	○	Foundational models:	○	External models:	●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION — MITIGATION CONTROLS

MODEL 7.2

Model assets leak

Adversaries may target ML artifacts for exfiltration or as a basis for staging ML attacks. These artifacts encompass models, datasets and metadata generated during interactions with a model. Additionally, insiders risk leaking critical model assets like notebooks, features, model files, plots and metrics. Such leaks can expose trade secrets and sensitive organizational information, underlining the need for stringent security measures to protect these valuable assets.

Model operations →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources
- DASF 24 Control access to models and model assets
- DASF 33 Manage credentials securely to prevent credentials of data sources used for model training from leaking through models
- DASF 42 Employ data-centric MLOps and LLMOps to maintain separate model lifecycle stages
- DASF 55 Monitor audit logs to track access to data, AI and other resources

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

MODEL 7.3

ML supply chain vulnerabilities

Due to the extensive data, skills and computational resources required to train machine learning algorithms, it's common practice to reuse and slightly modify models developed by large corporations. For example, ResNet, a popular image recognition model from Microsoft, is often adapted for customer-specific tasks. These models are curated in a Model Zoo (Caffe hosts popular image recognition models) or hosted by third-party ML SaaS (OpenAI LLMs are an example). In this attack, the adversary attacks the models hosted in Caffe, thereby poisoning the well for anyone else. Adversaries can also host specialized models that will receive less scrutiny, akin to watering hole attacks.

Model operations →

- DASF 22 Build models with all representative, accurate and relevant data sources to minimize third-party dependencies for models and data where possible
- DASF 27 Pretrain a large language model (LLM) on your own IP
- DASF 42 Employ data-centric MLOps and LLMOps to promote models as code using CI/CD. Scan third-party models continuously to identify hidden cybersecurity risks and threats such as malware, vulnerabilities and integrity issues to detect possible signs of malicious activity, including malware, tampering and back doors.
- DASF 45 Evaluate models and validate (aka, stress testing) to verify
  reported function and disclosed weaknesses in the models
- DASF 48 Use hardened Runtime for Machine Learning
- DASF 53 Third-party library control
- DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	●

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL 7.4 **Source code control attack** The attacker might modify the source code used in the ML algorithm, such as the random number generator or any third-party libraries, which are often open source. Model operations →	DASF 52 Source code control to control and audit your knowledge object integrity DASF 53 Third-party library control for third-party library integrity DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

2.8 Model management

Responsible AI depends upon accountability. Accountability presupposes transparency. AI transparency reflects the extent to which information about an AI system and its outputs is available to individuals interacting with it — regardless of whether they're even aware that they're doing so.

Organizations can increase trust by creating a centralized AI inventor for model management: development, tracking, discovering, governing, encrypting and accessing models with proper security controls. Doing so reduces the risk of model theft, improper reuse and model inversion. Transparency is also added by appropriate levels of information based on the stage of the AI lifecycle and tailored to the role or knowledge of practitioners or individuals interacting with the AI system. By promoting higher levels of understanding, transparency increases confidence in the AI system.

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL MANAGEMENT 8.1 **Model attribution** Inadequate governance in machine learning, including a lack of robust access controls, unclear model classification and insufficient documentation, can lead to the improper use or sharing of models. This risk is particularly acute when transferring models outside their designed purpose. To mitigate these risks, groups that post models must provide precise descriptions of their intended use and document how they address potential risks. Model operations →	DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources DASF 28 Create model aliases, tags and annotations for documenting and discovering models DASF 29 Build MLOps workflows with human-in-the-loop (HITL) , model stage management and approvals DASF 51 Share data and AI assets securely

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION

MITIGATION CONTROLS

MODEL MANAGEMENT 8.2

Model theft

Training machine learning systems, particularly large language models, involves considerable investment. A significant risk is the potential theft of a system's knowledge through direct observation of their input and output observations, akin to reverse engineering. This can lead to unauthorized access, copying or exfiltration of proprietary models, resulting in economic losses, eroded competitive advantage and exposure of sensitive information.

This attack can be as simple as attackers making legitimate queries and analyzing the responses to recreate a model. Once replicated, the model can be inverted, enabling the attackers to extract feature information or infer details about the training data.

Model operations →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources
- DASF 24 Control access to models and model assets
- DASF 30 Encrypt models
- DASF 31 Secure model serving endpoints to prevent access and compute theft
- DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit APIs
- DASF 33 Manage credentials securely to prevent credentials of data sources used for model training from leaking through models
- DASF 51 Share data and AI assets securely
- DASF 55 Monitor audit logs
- DASF 59 Use clean rooms to collaborate in a secure environment

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

MODEL MANAGEMENT 8.3

Model lifecycle without HITL (human-in-the-loop)

Lack of sufficient controls in a machine learning and systems development lifecycle can result in the unintended deployment of incorrect or unapproved models to production. Implementing model lifecycle tracking within an MLOps framework is advisable to mitigate this risk. This approach should include human oversight, ensuring permissions, version control and proper approvals are in place before models are promoted to production. Such measures are crucial for maintaining ML system integrity, reliability and security.

Model operations →

- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources
- DASF 24 Control access to models and model assets
- DASF 28 Create model aliases, tags and annotations
- DASF 29 Build MLOps workflows with human-in-the-loop (HILP) with permissions, versions and approvals to promote models to production
- DASF 42 Data-centric MLOps and LLMOps promote models as code using CI/CD

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	○
Pretrained LLMs:	●	Foundational models:	○	External models:	○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL MANAGEMENT 8.4 **Model inversion** In machine learning models, private assets like training data, features and hyperparameters, which are typically confidential, can potentially be recovered by attackers through a process known as model inversion. This technique involves reconstructing private elements without direct access, compromising the model's security. Model inversion falls under the "Functional Extraction" category in the MITRE ATLAS framework, highlighting its relevance as a significant security threat. Model operations →	DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform DASF 2 Sync users and groups to inherit your organizational roles to access data DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources DASF 24 Control access to models and model assets DASF 30 Encrypt models DASF 31 Secure model serving endpoints DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit APIs DASF 55 Monitor audit logs to track access to data, AI and other resources

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

2.9 Model Serving and inference requests

Model Serving exposes your machine learning models as scalable REST API endpoints for inference and provides a highly available and low-latency service for deploying models. Deploying a fully trained machine learning model introduces significant risks, including adversarial inputs, data poisoning, privacy concerns, access control issues, model vulnerabilities and versioning challenges. Using third-party or SaaS models amplifies these risks and introduces further limitations like lack of customization, model mismatch, ownership concerns and data privacy risks. Careful evaluation and mitigation strategies are necessary to securely and responsibly deploy fully trained models in production.

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION

MITIGATION CONTROLS

MODEL SERVING — INFERENCE REQUESTS 9.1

Prompt inject

A direct prompt injection occurs when a user injects text that's intended to alter the behavior of the LLM. Malicious input, known as "model evasion" in the MITRE ATLAS framework, is a significant threat to machine learning systems. These risks manifest as "adversarial examples": inputs deliberately designed to deceive models. Attackers use direct prompt injections to bypass safeguards in order to create misinformation and cause reputational damage. Attackers may wish to extract the system prompt or reveal private information provided to the model in the context but not intended for unfiltered access by the user. Large language model (LLM) plug-ins are particularly vulnerable, as they're typically required to handle untrusted input and it's difficult to apply adequate application control. Attackers can exploit such vulnerabilities, with severe potential outcomes including remote code execution.

Model deployment and serving →

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data

DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 24 Control access to models and model assets

DASF 30 Encrypt models

DASF 31 Secure model serving endpoints

DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

Designing robust prompts can help mitigate attacks such as jailbreaking.

Implement gates between users/callers and the actual model by performing input validation post-processing on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.

DASF 37 Set up inference tables for monitoring and debugging prompts

DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs

DASF 54 Implement AI guardrails

DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RISK/DESCRIPTION

MITIGATION CONTROLS

MODEL SERVING — INFERENCE REQUESTS 9.2

Model inversion

Malicious actors can recover the private assets used in machine learning models, known as "functional extraction" in the MITRE ATLAS framework. This process includes reconstructing private training data, features and hyperparameters the attacker can't otherwise access. The attacker can also recover a functionally equivalent model by iteratively querying the model.

Model deployment and serving →

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data

DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 24 Control access to models and model assets

DASF 30 Encrypt models

DASF 31 Secure model serving endpoints

DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

Designing robust prompts can help mitigate attacks such as jailbreaking.

Implement gates between users/callers and the actual model by performing input validation post-processing on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.

Open source and commercial solutions provide a variety of modules including prompt and output scanners for various responsible AI or jailbreaking attacks.

DASF 37 Set up inference tables for monitoring and debugging model prompts

DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs

DASF 54 Implement AI guardrails

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RISK/DESCRIPTION | MITIGATION CONTROLS

MODEL SERVING — INFERENCE REQUESTS 9.3

Model breakout

Malicious users can exploit adversarial examples to mislead machine learning systems, including large language models (LLMs). These specially crafted inputs aim to disrupt the normal functioning of these systems, leading to several potential hazards. An attacker might use these examples to force the system to deviate from its intended environment, exfiltrate sensitive data or interact inappropriately with other systems. Additionally, adversarial inputs can cause false predictions, leak sensitive information from the training data, or manipulate the system into executing unintended actions on internal and external systems.

Model deployment and serving →

DASF 34 Run models in multiple layers of isolation with unprivileged VMs and network segregation. Protects back-end internal systems from LLM access. The most reliable mitigation is to always treat all LLM output as potentially malicious and remember that an untrusted entity has been able to inject text as user input. All LLM output should be inspected and sanitized before being further parsed to extract information related to the plug-in. Plug-in templates should be parameterized wherever possible, and any calls to external services must be strictly parameterized at all times and made in a least-privileged context.

DASF 37 Set up inference tables for monitoring and debugging model prompts

DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

MODEL SERVING — INFERENCE REQUESTS 9.4

Looped input

There is a notable risk in machine learning systems when the output produced by the system is reintroduced into the real world and subsequently cycles back as input, creating a harmful feedback loop. This can reinforce removing security filters, biases or errors, potentially leading to increasingly skewed or inaccurate model performance and unintended system behaviors.

Model deployment and serving →

DASF 37 Set up inference tables for monitoring and debugging models to capture incoming requests and outgoing responses to your model serving endpoint and automatically log them in tables. Afterward, you can use the data in this table to monitor, debug and improve ML models and decide if these inferences are of sufficient quality for input to model training.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RISK/DESCRIPTION

MODEL SERVING — INFERENCE REQUESTS 9.5

Infer training data membership

Adversaries may pose a significant privacy threat to machine learning systems by simulating or inferring whether specific data samples were part of a model's training set. Such inferences can be made by:

- Using techniques like Train Proxy via Replication to create and host shadow models replicating the target model's behavior
- Analyzing the statistical patterns in the model's prediction scores to conclude the training data

These methods can lead to the unintended leakage of sensitive information, such as individuals' personally identifiable information (PII) in the training dataset or other forms of protected intellectual property.

Model deployment and serving →

MITIGATION CONTROLS

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources
- DASF 24 Control access to models and model assets
- DASF 28 Create model aliases, tags and annotations
- DASF 30 Encrypt models
- DASF 31 Secure model serving endpoints
- DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

  Designing robust prompts can help mitigate attacks such as jailbreaking.

  Implement gates between users/callers and the actual model by performing input validation post-processing on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.
- DASF 37 Set up inference tables for monitoring and debugging models
- DASF 45 Evaluate models for custom evaluation metrics
- DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs
- DASF 54 Implement AI guardrails
- DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RISK/DESCRIPTION

MITIGATION CONTROLS

MODEL SERVING — INFERENCE REQUESTS 9.6

Discover ML model ontology

Adversaries may aim to uncover the ontology of a machine learning model's output space, such as identifying the range of objects or responses the model is designed to detect. This can be achieved through repeated queries to the model, which may force it to reveal its classification system or by accessing its configuration files or documentation. Understanding a model's ontology allows adversaries to gain insights in designing targeted attacks that exploit specific vulnerabilities or characteristics.

Model deployment and serving →

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data

DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 24 Control access to models and model assets

DASF 28 Create model aliases, tags and annotations

DASF 30 Encrypt models

DASF 31 Secure model serving endpoints

DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

Designing robust prompts can help mitigate attacks such as jailbreaking.

Implement gates between users/callers and the actual model by performing input validation post-processing on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.

Open source and commercial solutions provide a variety of modules including prompt and output scanners for various responsible AI or jailbreaking attacks.

DASF 37 Set up inference tables for monitoring and debugging models

DASF 45 Evaluate models for custom evaluation metrics

DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs

DASF 54 Implement AI guardrails

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

RISK/DESCRIPTION

MODEL SERVING — INFERENCE REQUESTS 9.7

Denial of service (DoS)

Adversaries may target machine learning systems with a flood of requests to degrade or shut down the service. Since many machine learning systems require significant amounts of specialized compute, they're often expensive bottlenecks that can become overloaded. Adversaries can intentionally craft inputs that require heavy amounts of useless compute from the machine learning system.

Model deployment and serving →

MITIGATION CONTROLS

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data

DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 24 Control access to models and model assets

DASF 30 Encrypt models

DASF 31 Secure model serving endpoints

DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

Designing robust prompts can help mitigate attacks such as jailbreaking.

Implement gates between users/callers and the actual model by performing input validation postprocessing on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.

DASF 37 Set up inference tables for monitoring and debugging prompts

DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION | **MITIGATION CONTROLS**

MODEL SERVING — INFERENCE REQUESTS 9.8

LLM hallucinations

Large language models (LLMs) are known to inadvertently generate incorrect, misleading or factually false outputs, or leak sensitive data. This situation may arise when training models on datasets containing potential biases in their training data, limitations in contextual understanding or confidential information.

Model deployment and serving →

- DASF 25 Use retrieval augmented generation (RAG) with large language models (LLMs)

and/or

- DASF 26 Fine-tune large language models (LLMs) on highly relevant, contextual data to reduce the risks of LLMs by grounding with the domain-specific data
- DASF 27 Pretrain a large language model (LLM) on highly relevant, contextual data to reduce the risks of LLMs by grounding with the domain-specific data. The LLMs will investigate that data for giving the responses.
- DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs
- DASF 49 Automate LLM evaluation to evaluate RAG applications with LLM-as-a-judge and get out-of-the-box metrics like toxicity, latency, tokens and more to quickly and efficiently compare and contrast various LLMs to navigate your RAG application requirements
- DASF 54 Implement AI guardrails

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	○	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	●

MODEL SERVING — INFERENCE REQUESTS 9.9

Input resource control

The attacker might modify or exfiltrate resources (e.g., documents, web pages) that will be ingested by the GenAI model at runtime via the RAG process. This capability is used for indirect prompt injection attacks. For example, rows from a database or text from a PDF document that are intended to be summarized generically by the LLM can be extracted by simply asking for them via direct prompt injection.

Model deployment and serving →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources that are used for RAG
- DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs
- DASF 54 Implement AI guardrails
- DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses

Additional controls to consider:
Please see the companion compendium document (Google sheet, Excel) for a collection of third-party tools.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

RISK/DESCRIPTION | MITIGATION CONTROLS

MODEL SERVING — INFERENCE REQUESTS 9.10

Accidental exposure of unauthorized data to models

In GenAI, large language models (LLMs) are also becoming an integral part of the infrastructure and software applications. LLMs are being used to create more powerful online search, help software developers write code and even power chatbots that help with customer service. LLMs are being integrated with corporate databases and documents to enable powerful retrieval augmented generation (RAG) scenarios when LLMs are adapted to specific domains and use cases. For example: rows from a database or text from a PDF document that are intended to be summarized generically by the LLM. These scenarios in effect expose a new attack surface to potentially confidential and proprietary enterprise data that isn't sufficiently secured or overprivileged, which can lead to use of unauthorized data as an input source to models. A similar risk exists for tabular data models that rely upon lookups to feature store tables at inference time.

Model deployment and serving →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests
- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources
- DASF 16 Secure model features to reduce the risk of malicious actors manipulating the features that feed into ML training
- DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs
- DASF 55 Monitor audit logs
- DASF 57 Use attribute-based access controls (ABAC)
- DASF 58 Protect data with filters and masking
- DASF 59 Use clean rooms to collaborate in a secure environment

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

MODEL SERVING — INFERENCE REQUESTS 9.11

Model inference API access

Adversaries may gain access to a model via legitimate access to the inference API. Inference API access can be a source of information to the adversary (Discover ML Model Ontology, Discover ML Model Family), a means of staging the attack (Verify Attack, Craft Adversarial Data) or for introducing data to the target system for Impact (Evade ML Model, Erode ML Model Integrity).

Model deployment and serving →

- DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform
- DASF 2 Sync users and groups to inherit your organizational roles to access data
- DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform
- DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests
- DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources that are used for RAG
- DASF 31 Secure model serving endpoints
- DASF 33 Manage credentials securely to prevent credentials of data sources used for model training from leaking through models
- DASF 55 Monitor audit logs to track access to data, AI and other resources

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL SERVING — INFERENCE REQUESTS 9.12 **LLM jailbreak** An adversary may use a carefully crafted LLM prompt injection designed to place an LLM in a state in which it will freely respond to any user input, bypassing any controls, restrictions or guardrails placed on the LLM. Once successfully jailbroken, the LLM can be used in unintended ways by the adversary. Model deployment and serving →	DASF 1 SSO with IdP and MFA to authenticate and limit who can access your data and AI platform DASF 2 Sync users and groups to inherit your organizational roles to authorize access to data DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources DASF 31 Secure model serving endpoints DASF 33 Manage credentials securely to prevent credentials of data sources used for model training from leaking through models DASF 54 Implement AI guardrails DASF 55 Monitor audit logs to track access to data, AI and other resources DASF 64 Limit access from AI models and agents **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○
MODEL SERVING — INFERENCE REQUESTS 9.13 **Excessive agency** Generative AI systems may undertake actions outside of the developer's intent, organizational policy and/or legislative, regulatory and contractual requirements, leading to unintended consequences. This issue is facilitated by excessive permissions, excessive functionality, excessive autonomy, poorly defined operational parameters or granting the AI system and/or AI agents the ability to make decisions, access data sources and enterprise objects, or act on systems without human intervention or oversight. Model deployment and serving →	DASF 6 Classify data DASF 11 Capture and view data lineage DASF 55 Monitor audit logs to track access to data, AI and other resources DASF 57 Use attribute-based access controls (ABAC) DASF 58 Protect data with filters and masking DASF 62 Implement network segmentation DASF 64 Limit access from AI models and agents **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ● External models: ●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

2.10 Model Serving and inference response

While the technical intricacies of the algorithm may seem like the most vulnerable point for malicious actors seeking to compromise the integrity and reliability of the ML system, an equally effective, and often overlooked, attack vector lies in how it generates output (inference response). The inference response represents the real-world manifestation of the model's learned knowledge and forms the basis for its decision-making capabilities. Consequently, compromising the inference response directly can have devastating consequences, undermining the system's integrity and reliability.

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL SERVING — INFERENCE RESPONSE 10.1 **Lack of audit and monitoring inference quality** Effectively audit, track and assess the performance of machine learning models by monitoring inference tables to gain valuable insights into the model's decision-making process and identify any discrepancies or anomalies. These tables should include the model's user or system making the request, inputs and the corresponding predictions or outputs. Secure these tables with proper access controls. Monitoring the model serving endpoints provides real-time audit in operational settings. Model deployment and serving →	DASF 35 Track model performance to evaluate quality DASF 36 Set up monitoring alerts DASF 37 Set up inference tables for monitoring and debugging models to capture incoming requests and outgoing responses to your model serving endpoint and log them in a table. Afterward, you can use the data in this table to monitor, debug and improve ML models and decide if these inferences are of quality to use as input to model training. DASF 55 Monitor audit logs **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ● External models: ●
MODEL SERVING — INFERENCE RESPONSE 10.2 **Output manipulation** An attacker can compromise a machine learning system by tweaking its output stream, also known as a man-in-the-middle attack. This is achieved by intercepting the data transmission between the model's endpoint, which generates its predictions or outputs, and the intended receiver of this information. Such an attack poses a severe security threat, allowing the attacker to read or alter the communicated results, potentially leading to data leakage, misinformation or misguided actions based on manipulated data. Model deployment and serving →	DASF 30 Encrypt models for model endpoints with encryption in transit DASF 31 Secure model serving endpoints DASF 32 Streamline the usage and management of various large language model (LLM) providers to rate-limit inference queries allowed by the model. Then audit, reproduce and make your models more compliant. DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ● External models: ●

RISK/DESCRIPTION

MODEL SERVING — INFERENCE RESPONSE 10.3

Discover ML model ontology

Adversaries may aim to uncover the ontology of a machine learning model's output space, such as identifying the range of objects or responses the model is designed to detect. This can be achieved through repeated queries to the model, which may force it to reveal its classification system or by accessing its configuration files or documentation. Understanding a model's ontology allows adversaries to gain insights in designing targeted attacks that exploit specific vulnerabilities or characteristics.

Model deployment and serving →

MITIGATION CONTROLS

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data to restrict IP addresses

DASF 3 IP access lists to restrict the IP addresses that can authenticate to Databricks

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Unity Catalog privileges and securable objects for permissions model across all data assets to protect data and sources

DASF 24 Control access to models and model assets

DASF 28 Create and model aliases, tags and annotations in Unity Catalog for documenting and discovering models

DASF 30 Encrypt models

DASF 31 Secure serving endpoint with Model Serving

DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

The most reliable mitigation is to always treat all LLM productions as potentially malicious and under the control of any entity that has been able to inject text into the LLM user's input.

Implement gates between users/callers and the actual model by performing input validation on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.

DASF 37 Set up inference tables for monitoring and debugging models

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the model serving

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RISK/DESCRIPTION

MITIGATION CONTROLS

MODEL SERVING — INFERENCE RESPONSE 10.4

Discover ML model family

Adversaries targeting machine learning systems may strive to identify the general family or type of the model in use. Attackers can obtain this information from documentation that describes the model or through analyzing responses from carefully constructed inputs. Knowledge of the model's family is crucial for crafting attacks tailored to exploit the identified weaknesses of the model.

Model deployment and serving →

DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to access data

DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 24 Control access to models and model assets

DASF 28 Create model aliases, tags and annotations

DASF 30 Encrypt models

DASF 31 Secure model serving endpoints

DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.

Designing robust prompts can help mitigate attacks such as jailbreaking.

Implement gates between users/callers and the actual model by performing input validation postprocessing on all proposed queries, rejecting anything not meeting the model's definition of input correctness, and returning only the minimum amount of information needed to be useful.

Open source and commercial solutions provide a variety of modules including prompt and output scanners for various responsible AI or jailbreaking attacks.

DASF 37 Set up inference tables for monitoring and debugging models

DASF 45 Evaluate models for custom evaluation metrics

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the Model Serving

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	○
Pretrained LLMs:	○	Foundational models:	○	External models:	○

MODEL SERVING — INFERENCE RESPONSE 10.5

Black box attacks

Public or compromised private Model Serving connectors (e.g., API interfaces) are vulnerable to black box attacks. Although black box attacks generally require more trial-and-error attempts (inferences), they're notable for requiring significantly less knowledge of the target system. Successful black box attacks quickly erode trust in enterprises serving the model connectors.

Model deployment and serving →

DASF 30 Encrypt models for model endpoints with encryption in transit

DASF 31 Secure model serving endpoints

DASF 32 Streamline the usage and management of various large language model (LLM) providers to rate-limit inference queries allowed by the model. Then audit, reproduce and make your models more compliant.

DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the Model Serving

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	○	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	○	External models:	○

RISK/DESCRIPTION	MITIGATION CONTROLS
MODEL SERVING — INFERENCE RESPONSE 10.6 **Sensitive data output from a model** Without proper guardrails, generative AI outputs can contain confidential and/or sensitive information included in the model's training dataset, RAG data sources or data residing in data sources that the AI system is connected to (e.g., through language model tools such as agents or plug-ins). Examples of such information include that which is covered under data protection laws and regulations (e.g., personally identifiable information, protected health information, cardholder data) and corporate secrets. Model deployment and serving →	DASF 31 Secure model serving endpoints DASF 32 Streamline the usage and management of various large language model (LLM) providers to rate-limit inference queries allowed by the model. Then audit, reproduce and make your models more compliant. DASF 37 Set up inference tables for monitoring and debugging models DASF 54 Implement AI guardrails DASF 58 Protect data with filters and masking DASF 60 Rate limit number of inference queries to control capacity and ensure availability of the Model Serving **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ○ External models: ○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

2.11 Machine learning and operations (MLOps)

MLOps is a useful approach for creating quality AI solutions. It's a core function of machine learning engineering, focused on streamlining the process of taking machine learning models to production and then maintaining and monitoring them. By adopting an MLOps approach, data scientists and machine learning engineers can collaborate and increase the pace of model development and production by implementing continuous integration and continuous deployment (CI/CD) practices with proper monitoring, validation and governance of ML models with a "security in the process" mindset. Organizations without MLOps will risk missing some of the controls we discussed above or not applying them consistently at scale to manage thousands of models.

RISK/DESCRIPTION	MITIGATION CONTROLS
OPERATIONS 11.1 **Lack of MLOps — repeatable enforced standards** Operationalizing an ML solution requires joining data from predictions, monitoring and feature tables with other relevant data. Duplicating data, moving AI assets and driving governance and tracking across these stages may represent roadblocks to practitioners who would rather shortcut security controls to deliver their solution. Many organizations will find that the simplest way to securely combine ML solutions, input data and feature tables is to leverage the same platform that manages other production data. An ML solution comprises data, code and models. These assets must be developed, tested (staging) and deployed (production). For each of these stages, we also need to operate within an execution environment. Security is an essential component of all MLOps lifecycle stages. It ensures the complete lifecycle meets the required standards by keeping the distinct execution environments — development, staging and production. Operations and platform →	DASF 42 Employ data-centric MLOps and LLMOps best practices: separate environments by workspace and schema, promote models with code, MLOps Stacks for repeatable ML infra across environments DASF 44 Triggering actions in response to a specific event to trigger automated jobs to keep human-in-the-loop (HITL) DASF 45 Evaluate models to capture performance insights for language models **Applicable AI deployment model:** Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ● Pretrained LLMs: ● Foundational models: ● External models: ●

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

2.12 Data and AI platform security

Abundant real-world evidence suggests that actual attackers use simple tactics to subvert ML-driven systems. The choice of platform used for building and deploying AI models can have inherent risks and rewards.

RISK/DESCRIPTION | **MITIGATION CONTROLS**

PLATFORM 12.1

Lack of vulnerability management

Detecting and promptly addressing software vulnerabilities in systems that support data and AI/ML operations is a critical responsibility for software and service providers. Attackers do not necessarily need to target AI/ML algorithms directly; compromising the layers underlying AI/ML systems is often easier. Therefore, adhering to traditional security threat mitigation practices, such as a secure software development lifecycle, is essential across all software layers.

Operations and platform →

DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management to build, deploy and monitor AI/ML models on a platform that takes responsibility seriously and shares remediation timeline commitments

DASF 63 Update software

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	○

PLATFORM 12.2

Lack of penetration testing, red teaming and bug bounty

Penetration testing and bug bounty programs are vital in securing software that supports data and AI/ML operations. Unlike in direct attacks on AI/ML algorithms, adversaries often target underlying software risks, such as the OWASP Top 10. These foundational software layers are generally more prone to attacks than the AI/ML components. AI red teaming, especially for LLMs, is an important component of developing and deploying models safely.

Penetration testing involves skilled experts actively seeking and exploiting weaknesses, mimicking real attack scenarios. Bug bounty programs encourage external ethical hackers to find and report vulnerabilities, rewarding them for their discoveries. This combination of internal and external security testing enhances overall system protection, safeguarding the integrity of AI/ML infrastructures against cyberthreats.

Operations and platform →

DASF 39 Platform security — Incident Reponse Team to build, deploy and monitor AI/ML models on a platform that takes responsibility seriously and shares remediation timeline commitments. A bug bounty program removes a barrier researchers face in working with Databricks.

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	○

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

RISK/DESCRIPTION

MITIGATION CONTROLS

PLATFORM 12.3

Lack of incident response

AI/ML applications are mission-critical for business. Your chosen platform vendor must address security issues in machine learning operations quickly and effectively. The program should combine automated monitoring with manual analysis to address general and ML-specific threats.

Operations and platform →

DASF 39 Platform security — Incident Response Team

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	○

PLATFORM 12.4

Unauthorized privileged access

A significant security threat in machine learning platforms arises from malicious internal actors, such as employees or contractors. These individuals might gain unauthorized access to private training data or ML models, posing a grave risk to the integrity and confidentiality of the assets. Such unauthorized access can lead to data breaches, leakage of sensitive or proprietary information, business process abuses and potential sabotage of the ML systems. Implementing stringent internal security measures and monitoring protocols is critical to mitigate insider risks from the platform vendor.

Operations and platform →

DASF 40 Platform security — internal access

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	○

PLATFORM 12.5

Poor security in the software development lifecycle

Software platform security is an important part of any progressive security program. ML hackers have shown that they don't need to know sophisticated AI/ML concepts to compromise a system. Hackers have busied themselves with exposing and exploiting bugs in a platform where AI is built, as those systems are well known to them. The security of AI depends on the platform's security.

Operations and platform →

DASF 41 Platform security — secure SDLC

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	○

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

RISK/DESCRIPTION

MITIGATION CONTROLS

PLATFORM 12.6

Lack of compliance

As AI applications become prevalent, they're increasingly subject to scrutiny and regulations, such as the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States. Navigating these regulations can be complex, particularly regarding data privacy and user rights. Utilizing a compliance-certified platform can be a significant advantage for organizations. These platforms are specifically designed to meet regulatory standards, providing essential tools and resources to help organizations build and deploy AI applications that are compliant with these laws. By leveraging such platforms, organizations can more effectively address regulatory compliance challenges, ensuring their AI initiatives align with legal requirements and best practices for data protection.

Operations and platform →

DASF 50 Platform compliance to build on a compliant platform

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	○

PLATFORM 12.7

Initial access

The adversary is trying to gain access to the machine learning system.

The target system could be a network, mobile device or an edge device such as a sensor platform. The machine learning capabilities used by the system could be local with onboard or cloud-enabled ML capabilities.

Initial access consists of techniques that use various entry vectors to gain their initial foothold within the system.

Operations and platform →

DASF 1 SSO with IdP and MFA to authenticate and limit who can access your data and AI platform

DASF 2 Sync users and groups to inherit your organizational roles to authorize access to data

DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform

DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests

DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources

DASF 31 Secure model serving endpoints

DASF 55 Monitor audit logs to track access to data, AI and other resources

Applicable AI deployment model:

Predictive ML models:	●	RAG-LLMs:	●	Fine-tuned LLMs:	●
Pretrained LLMs:	●	Foundational models:	●	External models:	●

03

Understanding AI Risk Mitigation Controls

The recommended mitigation controls outlined below are designed to address the 62 technical security risks across the 12 AI components as highlighted earlier in this document. These controls are platform-agnostic and can be implemented on any data and AI platform, so you can integrate and map them into your existing infrastructure.

For organizations using Databricks, we've included references to knowledge articles that provide detailed guidance on implementing these mitigation controls within the Databricks ecosystem. These controls also map to the Databricks shared responsibility model (documentation for AWS, Azure or GCP). Please note that the Databricks control category column describes whether a control is:

- A simple configuration (e.g., a technical setting that implements a security capability)
- An implementation step (e.g., customizable settings that customize security controls to your risk posture)
- An out-of-the-box capability (enabled-by-default controls)
- Part of the Databricks Academy portal

You can learn more about the Databricks Data Intelligence Platform in the Appendix section of this document.

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

CONTROL/RISK

DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 1 SSO with IdP and MFA

RISKS

RAW DATA 1.1 · DATA PREP 2.1 · DATA PREP 2.2 · DATA PREP 2.3 · DATA PREP 2.4 · DATASETS 3.1 · EVALUATION 6.1 · MODEL 7.1 · MODEL 7.2 · MODEL MANAGEMENT 8.2 · MODEL MANAGEMENT 8.4 · MODEL SERVING – INFERENCE REQUESTS 9.1 · MODEL SERVING – INFERENCE REQUESTS 9.2 · MODEL SERVING – INFERENCE REQUESTS 9.5 · MODEL SERVING – INFERENCE REQUESTS 9.6 · MODEL SERVING – INFERENCE REQUESTS 9.7 · MODEL SERVING – INFERENCE REQUESTS 9.9 · MODEL SERVING – INFERENCE REQUESTS 9.10 · MODEL SERVING – INFERENCE REQUESTS 9.11 · MODEL SERVING – INFERENCE REQUESTS 9.12 · MODEL SERVING – INFERENCE RESPONSE 10.3 · MODEL SERVING – INFERENCE RESPONSE 10.4 · PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Implementing single sign-on with an identity provider's (IdP) multifactor authentication is critical for secure authentication. It adds an extra layer of security, ensuring that only authorized users access the Databricks Platform.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 2 Sync users and groups

RISKS

RAW DATA 1.1 · DATA PREP 2.1 · DATA PREP 2.2 · DATA PREP 2.3 · DATA PREP 2.4 · DATASETS 3.1 · EVALUATION 6.1 · MODEL 7.2 · MODEL MANAGEMENT 8.2 · MODEL MANAGEMENT 8.4 · MODEL SERVING – INFERENCE REQUESTS 9.1 · MODEL SERVING – INFERENCE REQUESTS 9.2 · MODEL SERVING – INFERENCE REQUESTS 9.5 · MODEL SERVING – INFERENCE REQUESTS 9.6 · MODEL SERVING – INFERENCE REQUESTS 9.7 · MODEL SERVING – INFERENCE REQUESTS 9.9 · MODEL SERVING – INFERENCE REQUESTS 9.10 · MODEL SERVING – INFERENCE REQUESTS 9.11 · MODEL SERVING – INFERENCE REQUESTS 9.12 · MODEL SERVING – INFERENCE RESPONSE 10.3 · MODEL SERVING – INFERENCE RESPONSE 10.4 · PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Synchronizing users and groups from your identity provider (IdP) with Databricks using the SCIM standard facilitates consistent and automated user provisioning for enhancing security.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 3 Restrict access using IP access lists

RISKS

- RAW DATA 1.1
- DATA PREP 2.1
- DATA PREP 2.2
- DATA PREP 2.3
- DATA PREP 2.4
- DATASETS 3.1
- EVALUATION 6.1
- MODEL 7.2
- MODEL MANAGEMENT 8.2
- MODEL MANAGEMENT 8.4
- MODEL SERVING — INFERENCE REQUESTS 9.1
- MODEL SERVING — INFERENCE REQUESTS 9.2
- MODEL SERVING — INFERENCE REQUESTS 9.5
- MODEL SERVING — INFERENCE REQUESTS 9.6
- MODEL SERVING — INFERENCE REQUESTS 9.7
- MODEL SERVING — INFERENCE REQUESTS 9.9
- MODEL SERVING — INFERENCE REQUESTS 9.10
- MODEL SERVING — INFERENCE REQUESTS 9.11
- MODEL SERVING — INFERENCE REQUESTS 9.12
- MODEL SERVING — INFERENCE RESPONSE 10.3
- MODEL SERVING — INFERENCE RESPONSE 10.4
- PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Configure IP access lists to restrict authentication to Databricks from specific IP ranges, such as VPNs or office networks, and strengthen network security by preventing unauthorized access from untrusted locations.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 4 Restrict access using private link

RISKS

- RAW DATA 1.1
- DATA PREP 2.1
- DATA PREP 2.2
- DATA PREP 2.3
- DATA PREP 2.4
- DATASETS 3.1
- EVALUATION 6.1
- MODEL 7.2
- MODEL MANAGEMENT 8.2
- MODEL MANAGEMENT 8.4
- MODEL SERVING — INFERENCE REQUESTS 9.1
- MODEL SERVING — INFERENCE REQUESTS 9.2
- MODEL SERVING — INFERENCE REQUESTS 9.5
- MODEL SERVING — INFERENCE REQUESTS 9.6
- MODEL SERVING — INFERENCE REQUESTS 9.7
- MODEL SERVING — INFERENCE REQUESTS 9.9
- MODEL SERVING — INFERENCE REQUESTS 9.10
- MODEL SERVING — INFERENCE REQUESTS 9.11
- MODEL SERVING — INFERENCE REQUESTS 9.12
- MODEL SERVING — INFERENCE RESPONSE 10.3
- MODEL SERVING — INFERENCE RESPONSE 10.4
- PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Use AWS PrivateLink, Azure Private Link or GCP Private Service Connect to create a private network route between the customer and the Databricks control plane or the control plane and the customer's compute plane environments to enhance data security by avoiding public internet exposure.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 5 Control access to data and other objects

RISKS

RAW DATA 1.1 · RAW DATA 1.4 · RAW DATA 1.11 · DATA PREP 2.1 · DATASETS 3.1 · DATASETS 3.2 · DATASETS 3.3 · GOVERNANCE 4.1 · EVALUATION 6.1 · MODEL 7.1 · MODEL 7.2 · MODEL MANAGEMENT 8.1 · MODEL MANAGEMENT 8.2 · MODEL MANAGEMENT 8.3 · MODEL MANAGEMENT 8.4 · MODEL SERVING — INFERENCE REQUESTS 9.1 · MODEL SERVING — INFERENCE REQUESTS 9.2 · MODEL SERVING — INFERENCE REQUESTS 9.5 · MODEL SERVING — INFERENCE REQUESTS 9.6 · MODEL SERVING — INFERENCE REQUESTS 9.7 · MODEL SERVING — INFERENCE REQUESTS 9.9 · MODEL SERVING — INFERENCE REQUESTS 9.10 · MODEL SERVING — INFERENCE REQUESTS 9.11 · MODEL SERVING — INFERENCE REQUESTS 9.12 · MODEL SERVING — INFERENCE RESPONSE 10.3 · MODEL SERVING — INFERENCE RESPONSE 10.4 · PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Implementing Unity Catalog for unified permissions management and assets simplifies access control and enhances security.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 6 Classify data

RISKS

RAW DATA 1.2 · MODEL SERVING — INFERENCE REQUESTS 9.13

DESCRIPTION

Tags are attributes containing keys and optional values that you can apply to different securable objects in Unity Catalog. Organizing securable objects with tags in Unity Catalog aids in efficient data management, data discovery and classification, essential for handling large datasets.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 7 Enforce data quality checks on batch and streaming datasets

RISKS

RAW DATA 1.3 · RAW DATA 1.9 · RAW DATA 1.11 · DATA PREP 2.1 · DATASETS 3.1 · GOVERNANCE 4.1 · EVALUATION 6.1

DESCRIPTION

Databricks Delta Live Tables (DLT) simplifies ETL development with declarative pipelines that integrate quality control checks and performance monitoring.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 8 Encrypt data at rest

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
RAW DATA 1.4 DATASETS 3.2 DATASETS 3.3	Databricks supports customer-managed encryption keys to strengthen data at rest protection and greater access control.	Configuration	AWS Azure GCP

DASF 9 Encrypt data in transit

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
RAW DATA 1.4 DATASETS 3.2 DATASETS 3.3	Databricks supports TLS 1.2+ encryption to protect customer data during transit. This applies to data transfer between the customer and the Databricks control plane and within the compute plane. Customers can also secure inter-cluster communications within the compute plane per their security requirements.	Out-of-the-box	AWS Azure GCP

DASF 10 Version data

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
RAW DATA 1.5 RAW DATA 1.7	Store data in lakehouse architecture using Delta tables. Delta tables can be versioned to revert any user or malicious actor poisoning of data. Data can be stored in lakehouse architecture in the customer's cloud account. Both raw data and feature tables are stored as Delta tables with access controls to determine who can read and modify them. Data lineage with Unity Catalog helps track and audit changes and the origin of ML data sources. Each operation that modifies a Delta Lake table creates a new table version. User actions are tracked and audited, and lineage of transformations is available all in the same platform. You can use history information to audit operations, roll back a table or query a table at a specific point in time using time travel.	Implementation	AWS Azure GCP

DASF 11 Capture and view data lineage

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
RAW DATA 1.6 RAW DATA 1.11 DATA PREP 2.1 DATASETS 3.1 GOVERNANCE 4.1 EVALUATION 6.1 MODEL SERVING — INFERENCE REQUESTS 9.13	Unity Catalog tracks and visualizes real-time data lineage across all languages to the column level, providing a traceable history of an object from notebooks, workflows, models and dashboards. This enhances transparency and compliance, with accessibility provided through the Catalog Explorer.	Out-of-the-box	AWS Azure GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 12 Delete records from datasets

RISKS	DESCRIPTION	CONTROL CATEGORY
RAW DATA 1.8	Data governance in Delta Lake, the lakehouse storage layer, utilizes its atomicity, consistency, isolation, durability (ACID) properties for effective data management. This includes the capability to remove data based on specific predicates from a Delta Table, including the complete removal of data's history, supporting compliance with regulations like GDPR and CCPA.	Implementation PRODUCT REFERENCE AWS | Azure | GCP

DASF 13 Use near real-time data

RISKS	DESCRIPTION	CONTROL CATEGORY
RAW DATA 1.9	Use Databricks for near real-time data ingestion, processing, machine learning and AI for streaming data.	Implementation PRODUCT REFERENCE AWS | Azure | GCP

DASF 14 Audit actions performed on datasets

RISKS	DESCRIPTION	CONTROL CATEGORY
RAW DATA 1.10 DATASETS 3.1	Databricks auditing, enhanced by Unity Catalog events, delivers fine-grained visibility into data access and user activities. This is vital for robust data governance and security, especially in regulated industries. It enables organizations to proactively identify and manage overentitled users, enhancing data security and ensuring compliance.	Implementation PRODUCT REFERENCE AWS | Azure | GCP

DASF 15 Explore datasets and identify problems

RISKS	DESCRIPTION	CONTROL CATEGORY
DATA PREP 2.1	Iteratively explore, share and prep data for the machine learning lifecycle by creating reproducible, editable and shareable datasets, tables and visualizations. Within Databricks this EDA process can be accelerated with Mosaic AI AutoML. AutoML not only generates baseline models given a dataset, but also provides the underlying model training code in the form of a Python notebook. Notably for EDA, AutoML calculates summary statistics on the provided dataset, creating a notebook for the data scientist to review and adapt.	Implementation PRODUCT REFERENCE AWS | Azure | GCP

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 16 Secure model features

RISKS

DATA PREP 2.1 | DATA PREP 2.2 | DATASETS 3.1 | GOVERNANCE 4.1 | ALGORITHMS 5.2 | MODEL SERVING — INFERENCE REQUESTS 9.10

DESCRIPTION

Databricks Feature Store is a centralized repository that enables data scientists to find and share features and also ensures that the same code used to compute the feature values is used for model training and inference. Unity Catalog's capabilities, such as security, lineage, table history, tagging and cross-workspace access, are automatically available to the feature table to reduce the risk of malicious actors manipulating the features that feed into ML training.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 17 Track and reproduce the training data used for ML model training

RISKS

RAW DATA 1.11 | DATA PREP 2.4 | DATASETS 3.1 | GOVERNANCE 4.1 | ALGORITHMS 5.2 | MODEL SERVING — INFERENCE REQUESTS 9.11

DESCRIPTION

MLflow with Delta Lake tracks the training data used for ML model training. It also enables the identification of specific ML models and runs derived from particular datasets for regulatory and auditable attribution.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 18 Govern model assets

RISKS

GOVERNANCE 4.1

DESCRIPTION

With Unity Catalog, organizations can implement a unified governance framework for their structured and unstructured data, machine learning models, notebooks, features, functions and files, enhancing security and compliance across clouds and platforms.

Maintain an updated inventory of high-impact AI use cases, including details on purpose, benefits, risks and risk management practices, utilizing Unity Catalog to document data assets. This centralized platform enables tagging, describing and managing metadata, allowing users to easily discover and understand data through a search interface. Unity Catalog is particularly useful for scenarios where multiple teams need to access and utilize data across different workspaces while maintaining data governance and security through access controls.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 19 Manage end-to-end machine learning lifecycle

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
GOVERNANCE 4.2 MODEL 7.1	Databricks includes a managed version of MLflow featuring enterprise security controls and high availability. It supports functionalities like experiments, run management and notebook revision capture. MLflow on Databricks allows tracking and measuring machine learning model training runs, logging model training artifacts and securing machine learning projects.	Implementation	AWS Azure GCP

DASF 20 Track ML training runs

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
ALGORITHMS 5.1 ALGORITHMS 5.3	MLflow tracking facilitates the automated recording and retrieval of experiment details, including algorithms, code, datasets, parameters, configurations, signatures and artifacts.	Implementation	AWS Azure GCP

DASF 21 Monitor data and AI system from a single pane of glass

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
RAW DATA 1.3 GOVERNANCE 4.2 ALGORITHMS 5.2	Databricks Lakehouse Monitoring offers a single pane of glass to centrally track tables' data quality and statistical properties and automatically classifies data. It can also track the performance of machine learning models and model serving endpoints by monitoring inference tables containing model inputs and predictions through a single pane of glass.	Implementation	AWS Azure N/A

DASF 22 Build models with all representative, accurate and relevant data sources

RISKS	DESCRIPTION	CONTROL CATEGORY	PRODUCT REFERENCE
EVALUATION 6.2 MODEL 7.3	Harnessing internal data and intellectual property to customize large AI models can offer a significant competitive edge. However, this process can be complex, involving coordination across various parts of the organization. The Data Intelligence Platform addresses this challenge by integrating data across traditionally isolated departments and systems. This integration facilitates a more cohesive data and AI strategy, enabling the effective training, testing and evaluation of models using a comprehensive dataset. Use caution when preparing data for traditional models and GenAI training to ensure that you are not unintentionally including data that causes legal conflicts, such as copyright violations, privacy violations or HIPAA violations.	Implementation	AWS Azure GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 23 Register, version, approve, promote and deploy model

RISKS

- MODEL 7.1

DESCRIPTION

MLflow Model Registry supports managing the machine learning model lifecycle with capabilities for lineage tracking, versioning, staging and model serving.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 24 Control access to models and model assets

RISKS

- MODEL 7.2
- MODEL MANAGEMENT 8.2
- MODEL MANAGEMENT 8.3
- MODEL MANAGEMENT 8.4
- MODEL SERVING – INFERENCE REQUESTS 9.1
- MODEL SERVING – INFERENCE REQUESTS 9.2
- MODEL SERVING – INFERENCE REQUESTS 9.5
- MODEL SERVING – INFERENCE REQUESTS 9.6
- MODEL SERVING – INFERENCE REQUESTS 9.7
- MODEL SERVING – INFERENCE RESPONSE 10.3
- MODEL SERVING – INFERENCE RESPONSE 10.4

DESCRIPTION

Organizations commonly encounter challenges in tracking and controlling access to ML models, auditing their usage and understanding their evolution in complex machine learning workflows. Unity Catalog integrates with the MLflow Model Registry across model lifecycles. This approach simplifies the management and oversight of ML models, proving particularly valuable in environments with multiple teams and diverse projects.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 25 Use retrieval augmented generation (RAG) with large language models (LLMs)

RISKS

- EVALUATION 6.2
- MODEL SERVING – INFERENCE REQUESTS 9.8

DESCRIPTION

Generating relevant and accurate responses in large language models (LLMs) while avoiding hallucinations requires grounding them in domain-specific knowledge. Retrieval augmented generation (RAG) addresses this by breaking down extensive datasets into manageable segments (“chunks”) that are “vector embedded.” These vector embeddings are mathematical representations that help the model understand and quantify different data segments. As a result, LLMs produce responses that are contextually relevant and deeply rooted in the specific domain knowledge.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 26 Fine-tune large language models (LLMs)

RISKS

- MODEL SERVING – INFERENCE REQUESTS 9.8

DESCRIPTION

Data is your competitive advantage. Use it to customize large AI models to beat your competition. Produce new model variants with tailored LLM response style and structure via fine-tuning.

Fine-tune your own LLM with open models to own your IP.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

DASF 27 Pretrain a large language model (LLM)

RISKS

- RAW DATA 1.8
- MODEL 7.3
- MODEL SERVING — INFERENCE REQUESTS 9.8

DESCRIPTION

Data is your competitive advantage. Use it to customize large AI models to beat your competition by pretraining models with your data, imbuing the model with domain-specific knowledge, vocabulary and semantics. Pretrain your own LLM with MosaicML to own your IP.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 28 Create model aliases, tags and annotations

RISKS

- MODEL MANAGEMENT 8.1
- MODEL MANAGEMENT 8.3
- MODEL SERVING — INFERENCE REQUESTS 9.5
- MODEL SERVING — INFERENCE REQUESTS 9.6
- MODEL SERVING — INFERENCE RESPONSE 10.3
- MODEL SERVING — INFERENCE RESPONSE 10.4

DESCRIPTION

Model aliases in machine learning workflows allow you to assign a mutable, named reference to a specific version of a registered model. This functionality is beneficial for tracking and managing different stages of a model's lifecycle, indicating the current deployment status of any given model version.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 29 Build MLOps workflows

RISKS

- RAW DATA 1.8
- MODEL MANAGEMENT 8.1
- MODEL MANAGEMENT 8.3

DESCRIPTION

The lakehouse forms the foundation of a data-centric AI platform. Key to this is the ability to manage both data and AI assets from a unified governance solution on the lakehouse. Databricks Unity Catalog enables this by providing centralized access control, auditing, approvals, model workflow, lineage and data discovery capabilities across Databricks workspaces.

These benefits are now extended to MLflow models with the introduction of models in Unity Catalog. Through providing a hosted version of the MLflow Model Registry in Unity Catalog, the full lifecycle of an ML model can be managed while leveraging Unity Catalog's capability to share assets across Databricks workspaces and trace lineage across both data and models.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

DASF 30 Encrypt models

RISKS

- MODEL MANAGEMENT 8.2
- MODEL MANAGEMENT 8.4
- MODEL SERVING — INFERENCE REQUESTS 9.1
- MODEL SERVING — INFERENCE REQUESTS 9.2
- MODEL SERVING — INFERENCE REQUESTS 9.5
- MODEL SERVING — INFERENCE REQUESTS 9.6
- MODEL SERVING — INFERENCE REQUESTS 9.7
- MODEL SERVING — INFERENCE RESPONSE 10.2
- MODEL SERVING — INFERENCE RESPONSE 10.3
- MODEL SERVING — INFERENCE RESPONSE 10.4
- MODEL SERVING — INFERENCE RESPONSE 10.5

DESCRIPTION

The Databricks Platform secures model assets and their transfer with TLS 1.2+ in-transit encryption. Additionally, Unity Catalog's managed model registry provides encryption at rest for persisting models, further enhancing security.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 31 Secure model serving endpoints

RISKS

- MODEL MANAGEMENT 8.2
- MODEL MANAGEMENT 8.4
- MODEL SERVING — INFERENCE REQUESTS 9.1
- MODEL SERVING — INFERENCE REQUESTS 9.2
- MODEL SERVING — INFERENCE REQUESTS 9.5
- MODEL SERVING — INFERENCE REQUESTS 9.6
- MODEL SERVING — INFERENCE REQUESTS 9.7
- MODEL SERVING — INFERENCE REQUESTS 9.11
- MODEL SERVING — INFERENCE REQUESTS 9.12
- MODEL SERVING — INFERENCE RESPONSE 10.2
- MODEL SERVING — INFERENCE RESPONSE 10.3
- MODEL SERVING — INFERENCE RESPONSE 10.4
- MODEL SERVING — INFERENCE RESPONSE 10.5
- MODEL SERVING — INFERENCE RESPONSE 10.6
- PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Manage permissions on your Model Serving endpoints. Model serving involves risks of unauthorized data access and model tampering, which can compromise the integrity and reliability of machine learning deployments. Mosaic AI Model Serving addresses these concerns by providing secure-by-default REST API endpoints for MLflow machine learning models, featuring autoscaling, high availability and low latency.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

DASF 32 Streamline the usage and management of various large language model (LLM) providers

RISKS

- MODEL MANAGEMENT 8.2
- MODEL MANAGEMENT 8.4
- MODEL SERVING – INFERENCE REQUESTS 9.1
- MODEL SERVING – INFERENCE REQUESTS 9.2
- MODEL SERVING – INFERENCE REQUESTS 9.5
- MODEL SERVING – INFERENCE REQUESTS 9.6
- MODEL SERVING – INFERENCE REQUESTS 9.7
- MODEL SERVING – INFERENCE RESPONSE 10.2
- MODEL SERVING – INFERENCE RESPONSE 10.3
- MODEL SERVING – INFERENCE RESPONSE 10.4
- MODEL SERVING – INFERENCE RESPONSE 10.5
- MODEL SERVING – INFERENCE RESPONSE 10.6

DESCRIPTION

External models are third-party models hosted outside of Databricks. Supported by Model Serving AI Gateway, Databricks external models via the AI Gateway allow you to streamline the usage and management of various large language model (LLM) providers, such as OpenAI and Anthropic, within an organization. You can also use Mosaic AI Model Serving as a provider to serve predictive ML models, which offers rate limits for those endpoints. As part of this support, Model Serving offers a high-level interface that simplifies the interaction with these services by providing a unified endpoint to handle specific LLM-related requests. In addition, Databricks support for external models provides centralized credential management. By storing API keys in one secure location, organizations can enhance their security posture by minimizing the exposure of sensitive API keys throughout the system. It also helps to prevent exposing these keys within code or requiring end users to manage keys safely.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 33 Manage credentials securely

RISKS

- MODEL 7.2
- MODEL MANAGEMENT 8.2
- MODEL SERVING – INFERENCE REQUESTS 9.11
- MODEL SERVING – INFERENCE REQUESTS 9.12

DESCRIPTION

Databricks Secrets stores your credentials and references them in notebooks, scripts, configuration properties and jobs.

Integrating with heterogeneous systems requires managing a potentially large set of credentials and safely distributing them across an organization. Instead of directly entering your credentials into a notebook, use Databricks Secrets to store your credentials and reference them in notebooks and jobs to prevent credential leaks through models. Databricks secret management allows users to use and share credentials within Databricks securely. You can also choose to use a third-party secret management service, such as AWS Secrets Manager or a third-party secret manager.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 34 Run models in multiple layers of isolation

RISKS

- MODEL 7.1
- MODEL SERVING – INFERENCE REQUESTS 9.3

DESCRIPTION

Databricks serverless compute provides a secure-by-design model serving service featuring defense-in-depth controls like dedicated VMs, network segmentation and encryption for data in transit and at rest. It adheres to the principle of least privilege for enhanced security.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

Executive Summary

Introduction

Risks in AI System Components

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

Appendix: Understanding the Databricks Data Intelligence Platform

Appendix: Glossary

License

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 35 Track model performance

RISKS

- EVALUATION 6.3
- MODEL SERVING — INFERENCE RESPONSE 10.1

DESCRIPTION

Databricks Lakehouse Monitoring provides performance metrics and data quality statistics across all account tables. It tracks the performance of machine learning models and model serving endpoints by observing inference tables with model inputs and predictions.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 36 Set up monitoring alerts

RISKS

- RAW DATA 1.3
- MODEL SERVING — INFERENCE RESPONSE 10.1

DESCRIPTION

Databricks SQL alerts can monitor the metrics table for security-based conditions, ensuring data integrity and timely response to potential issues:

- **Statistic range Alert:** Triggers when a specific statistic, such as the fraction of missing values, exceeds a predetermined threshold
- **Data distribution shift alert:** Activates upon shifts in data distribution, as indicated by the drift metrics table
- **Baseline divergence alert:** Alerts if data significantly diverges from a baseline, suggesting potential needs for data analysis or model retraining, particularly in InferenceLog analysis

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 37 Set up inference tables for monitoring and debugging models

RISKS

- EVALUATION 6.3
- MODEL SERVING — INFERENCE REQUESTS 9.1
- MODEL SERVING — INFERENCE REQUESTS 9.2
- MODEL SERVING — INFERENCE REQUESTS 9.3
- MODEL SERVING — INFERENCE REQUESTS 9.4
- MODEL SERVING — INFERENCE REQUESTS 9.5
- MODEL SERVING — INFERENCE REQUESTS 9.6
- MODEL SERVING — INFERENCE REQUESTS 9.7
- MODEL SERVING — INFERENCE RESPONSE 10.1
- MODEL SERVING — INFERENCE RESPONSE 10.3
- MODEL SERVING — INFERENCE RESPONSE 10.4
- MODEL SERVING — INFERENCE RESPONSE 10.6

DESCRIPTION

Databricks inference tables automatically record incoming requests and outgoing responses to model serving endpoints, storing them as a Unity Catalog Delta table. This table can be used to monitor, debug and enhance ML models. By coupling inference tables with Lakehouse Monitoring, customers can also set up automated monitoring jobs and alerts on inference tables, such as monitoring text quality or toxicity from endpoints serving LLMs, etc.

Critical applications of an inference table include:

- **Retraining dataset creation:** Building datasets for the next iteration of your models
- **Quality monitoring:** Keeping track of production data and model performance
- **Diagnostics and debugging:** Investigating and resolving issues with suspicious inferences
- **Mislabeled data identification:** Compiling data that needs relabeling

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management

RISKS	DESCRIPTION	CONTROL CATEGORY
PLATFORM 12.1	Mitigating attacks on infrastructure hosting AI services, including AI red teaming for large language models, is crucial for safe model development and deployment. Regular security and penetration testing help identify and address infrastructure vulnerabilities before attackers can exploit them. Databricks operates a formal, documented vulnerability management program overseen by the chief security officer (CSO). The program is management approved, reviewed annually and communicated to all relevant internal parties. AI red teaming, especially for large language models, is an essential component of ensuring model safety and security. Databricks conducts regular AI red teaming on models and systems developed internally.	Out-of-the-box PRODUCT REFERENCE AWS | Azure | GCP

DASF 39 Platform security — Incident Response Team

RISKS	DESCRIPTION	CONTROL CATEGORY
PLATFORM 12.2 PLATFORM 12.3	Databricks has established a formal incident response plan that outlines key elements such as roles, responsibilities, escalation paths and external communication protocols. The platform handles over 9TB of audit logs daily, aiding customer and Databricks security investigations. A dedicated security incident response team operates an internal Databricks instance, consolidating essential log sources for thorough security analysis. Databricks ensures continual operational readiness with a 24/7/365 on-call rotation. Additionally, a proactive hunting program and a specialized detection team support the incident response program, require periodic AI audits and establish protocols for incident reporting, including logs review, performance monitoring and procedures to report and address misuse.	Out-of-the-box PRODUCT REFERENCE AWS | Azure | GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 40 Platform security — internal access

RISKS	DESCRIPTION	CONTROL CATEGORY
PLATFORM 12.4	Databricks personnel, by default, don't have access to customer workspaces or production environments. Access may be temporarily requested by Databricks staff for purposes such as investigating outages, security events or supporting deployments. Customers have the option to disable this access. Additionally, staff activity within these environments is recorded in customer audit logs. Accessing these areas requires multifactor authentication, and employees must connect to the Databricks VPN.	Out-of-the-box PRODUCT REFERENCE AWS | Azure | GCP

DASF 41 Platform security — secure SDLC

RISKS	DESCRIPTION	CONTROL CATEGORY
PLATFORM 12.5	Databricks engineering integrates security throughout the software development lifecycle (SDLC), encompassing both technical and process-level controls under the oversight of our chief security officer (CSO). Activities within our SDLC include: ▪ Code peer reviews ▪ Static and dynamic scans for code and containers, including dependencies ▪ Feature-level security reviews ▪ Annual software engineering security training ▪ Cross-organizational collaborations between security, product management, product security and security champions These development controls are augmented by internal and external penetration testing programs, with findings tracked for resolution and reported to our executive team. Databricks processes undergo an independent annual review, the results of which are published in our SOC 2 Type 2 report, available upon request.	Out-of-the-box PRODUCT REFERENCE AWS | Azure | GCP

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 42 Employ data-centric MLOps and LLMOps

RISKS

DATA PREP 2.1 DATA PREP 2.2 DATA PREP 2.3 DATA PREP 2.4 GOVERNANCE 4.2 ALGORITHMS 5.1 ALGORITHMS 5.3 EVALUATION 6.1 EVALUATION 6.3 MODEL 7.1 MODEL 7.2 MODEL 7.3 MODEL MANAGEMENT 8.3 OPERATIONS 11.1

DESCRIPTION

MLOps enhances efficiency, scalability, security and risk reduction in machine learning projects. Databricks integrates with MLflow, focusing on enterprise reliability, security and scalability for managing the machine learning lifecycle. The latest update to MLflow introduces new LLMOps features for better management and deployment of large language models (LLMs). This includes integrations with Hugging Face Transformers, OpenAI and the external models in Mosaic AI Model Serving.

MLflow also integrates with LangChain and a prompt engineering UI, facilitating generative AI application development for use cases such as chatbots, document summarization and text classification.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 43 Use access control lists

RISKS

DATA PREP 2.3 ALGORITHMS 5.3 MODEL 7.1

DESCRIPTION

Databricks access control lists (ACLs) enable you to configure permissions for accessing and interacting with workspace objects, including folders, notebooks, experiments, models, clusters, pools, jobs, Delta Live Tables pipelines, alerts, dashboards, queries and SQL warehouses.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 44 Triggering actions in response to a specific event

RISKS

EVALUATION 6.1 OPERATIONS 11.1

DESCRIPTION

Webhooks in the MLflow Model Registry enable you to automate machine learning workflow by triggering actions in response to specific events. These webhooks facilitate seamless integrations, allowing for the automatic execution of various processes. For example, webhooks are used for:

- **CI workflow trigger:** Validate your model automatically when creating a new version
- **Team notifications:** Send alerts through a messaging app when a model stage transition request is received
- **Model fairness evaluation:** Invoke a workflow to assess model fairness and bias upon a production transition request
- **Automated deployment:** Trigger a deployment pipeline when a new tag is created on a model

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 45 Evaluate models

RISKS

EVALUATION 6.1
EVALUATION 6.2
EVALUATION 6.3
MODEL 7.3
MODEL SERVING – INFERENCE REQUESTS 9.5
MODEL SERVING – INFERENCE REQUESTS 9.6
MODEL SERVING – INFERENCE RESPONSE 10.4
OPERATIONS 11.1

DESCRIPTION

Model evaluation is a critical component of the machine learning lifecycle. It provides data scientists with the tools to measure, interpret and explain the performance of their models. MLflow plays a critical role in accelerating model development by offering insights into the reasons behind a model's performance and guiding improvements and iterations. MLflow offers many industry-standard native evaluation metrics for classical machine learning algorithms and LLMs, and also facilitates the use of custom evaluation metrics.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 46 Store and retrieve embeddings securely

RISKS

MODEL SERVING – INFERENCE REQUESTS 9.1
MODEL SERVING – INFERENCE REQUESTS 9.2
MODEL SERVING – INFERENCE REQUESTS 9.5
MODEL SERVING – INFERENCE REQUESTS 9.6
MODEL SERVING – INFERENCE REQUESTS 9.7
MODEL SERVING – INFERENCE REQUESTS 9.8
MODEL SERVING – INFERENCE REQUESTS 9.9
MODEL SERVING – INFERENCE REQUESTS 9.10
MODEL SERVING – INFERENCE RESPONSE 10.4

DESCRIPTION

Mosaic AI Vector Search is a vector database that is built into the Databricks Data Intelligence Platform and integrated with its governance and productivity tools. A vector database is a database that is optimized to store and retrieve embeddings. Embeddings are mathematical representations of the semantic content of data, typically text or image data. Embeddings are usually generated by feature extraction models for text, image, audio or multimodal data, and are a key component of many GenAI applications that depend on finding documents or images that are similar to each other. Examples are RAG systems, recommender systems and image and video recognition.

Databricks implements the following security controls to protect your data:

- Every customer request to Vector S earch is logically isolated, authenticated and authorized
- Mosaic AI Vector Search encrypts all data at rest (AES-256) and in transit (TLS 1.2+)
- Optionally can be encrypted with customer-managed keys (CMK)

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 47 Compare LLM outputs on set prompts

RISKS

EVALUATION 6.2

DESCRIPTION

New, no-code visual tools allow users to compare models' output based on set prompts, which are automatically tracked within MLflow. With integration into Mosaic AI Model Serving, customers can deploy the best model to production. The AI Playground is a chat-like environment where you can test, prompt and compare LLMs.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

CONTROL/RISK

DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 48 Use hardened Runtime for Machine Learning

RISKS

MODEL 7.3

DESCRIPTION

Databricks Runtime for Machine Learning (Databricks Runtime ML) now automates cluster creation with versatile infrastructure, encompassing pre-built ML/DL libraries and custom library integration. Enhanced scalability and cost management tools optimize performance and expenditure. The refined user interface caters to various expertise levels, while new collaboration features support team-based projects. Comprehensive training resources and detailed documentation complement these improvements.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 49 Automate LLM evaluation

RISKS

EVALUATION 6.1

MODEL SERVING — INFERENCE REQUESTS 9.8

DESCRIPTION

The "LLM-as-a-judge" feature in MLflow 2.8 automates LLM evaluation, offering a practical alternative to human judgment. It's designed to be efficient and cost-effective, maintaining consistency with human scores. This tool supports various metrics, including standard and customizable GenAI metrics, and allows users to select an LLM as a judge and define specific grading criteria.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 50 Platform compliance

RISKS

PLATFORM 12.6

DESCRIPTION

Develop your solutions on a platform created using some of the most rigorous security and compliance standards in the world. Get independent audit reports verifying that Databricks adheres to security controls for ISO 27001, ISO 27018, SOC 1, SOC 2, FedRAMP, HITRUST, IRAP, etc.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 51 Share data and AI assets securely

RISKS

RAW DATA 1.1

RAW DATA 1.6

RAW DATA 1.7

DATASETS 3.1

MODEL MANAGEMENT 8.1

MODEL MANAGEMENT 8.2

DESCRIPTION

Databricks Delta Sharing lets you share data and AI assets securely in Databricks with users outside your organization, whether those users use Databricks or not.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 52 Source code control

RISKS

DATA PREP 2.1 MODEL 7.4

DESCRIPTION

Databricks Git Repository integration supports effective code and third-party libraries management, enhancing customer control over their development environment.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 53 Third-party library control

RISKS

ALGORITHMS 5.4 MODEL 7.3

MODEL 7.4

DESCRIPTION

The Databricks library management system allows administrators to manage the installation and usage of third-party libraries effectively. This feature enhances the security and efficiency of systems, pipelines and data by giving administrators precise control over their development environment.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 54 Implement AI guardrails

RISKS

MODEL SERVING — INFERENCE REQUESTS 9.1

MODEL SERVING — INFERENCE REQUESTS 9.2

MODEL SERVING — INFERENCE REQUESTS 9.5

MODEL SERVING — INFERENCE REQUESTS 9.6

MODEL SERVING — INFERENCE REQUESTS 9.8

MODEL SERVING — INFERENCE REQUESTS 9.9

MODEL SERVING — INFERENCE REQUESTS 9.12

MODEL SERVING — INFERENCE RESPONSE 10.6

DESCRIPTION

AI guardrails allow users to configure and enforce data compliance at the model serving endpoint level and to reduce harmful content on any requests sent to the underlying model. Bad requests and responses are blocked, and a default message is returned to the user. With AI guardrails, you can configure the following controls on your AI system:

- Safety filtering prevents your model from interacting with unsafe and harmful content, like violent crime, self-harm and hate speech
- Personally identifiable information (PII) detection to detect any sensitive information (such as names, addresses and credit card numbers) for users
- Topic moderation to list a set of allowed topics. Given a chat request, this guardrail flags the request if its topic isn't one of the permitted topics.
- Keyword filtering will specify different sets of invalid keywords for both the input and the output. One potential use case for keyword filtering is so the model doesn't talk about competitors.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

CONTROL/RISK

DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 55 Monitor audit logs

RISKS

RAW DATA 1.1 RAW DATA 1.10
DATA PREP 2.1 DATASETS 3.1
GOVERNANCE 4.1 ALGORITHMS 5.1
MODEL 7.1 MODEL 7.2
MODEL MANAGEMENT 8.2
MODEL MANAGEMENT 8.4
MODEL SERVING — INFERENCE REQUESTS 9.10
MODEL SERVING — INFERENCE REQUESTS 9.11
MODEL SERVING — INFERENCE REQUESTS 9.12
MODEL SERVING — INFERENCE REQUESTS 9.13
MODEL SERVING — INFERENCE RESPONSE 10.1
PLATFORM 12.7: INITIAL ACCESS

DESCRIPTION

Audit logs and system tables serve as a centralized operational data store, backed by Delta Lake and governed by Unity Catalog. Audit logs and system tables can be used for a variety of purposes, from user activity, model serving events and cost monitoring to audit logging. Databricks recommends that customers configure system tables and set up automated monitoring and alerting to meet their needs. The blog post Improve Lakehouse Security Monitoring Using System Tables in Databricks Unity Catalog is a good resource to help customers get started.

Customers that are using enhanced security monitoring or the compliance security profile can monitor and alert on suspicious activity detected by the behavior-based malware and file integrity monitoring agents.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | GCP

DASF 56 Restrict outbound connections from models

RISKS

MODEL 7.1 MODEL 7.3 MODEL 7.4
MODEL SERVING — INFERENCE REQUESTS 9.1
MODEL SERVING — INFERENCE REQUESTS 9.3
MODEL SERVING — INFERENCE REQUESTS 9.9

DESCRIPTION

Egress Control enables you to control outbound connections from your Model Serving compute resources.

With this feature, you can restrict access to the internet while allowing access via Unity Catalog Connections or Private Link. Further, this feature blocks direct access to cloud storage (over the shared S3 gateway) to ensure that all data access occurs via Unity Catalog–controlled paths to reduce the risk of data exfiltration.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS | Azure | N/A

DASF 57 Use attribute-based access controls (ABAC)

RISKS

MODEL SERVING — INFERENCE REQUESTS 9.10
MODEL SERVING — INFERENCE REQUESTS 9.13

DESCRIPTION

Attribute-based access controls (ABAC) allow data stewards to set policies on data and AI assets using various criteria like user-defined tags, workspace details, location, identity and time. Whether it's restricting sensitive data to authorized personnel or adjusting access dynamically based on project needs, ABAC ensures security measures are applied with detailed accuracy. Implement attribute-based access controls (ABAC) to define access policies based on attributes or characteristics of the user or the resource being accessed. Use row level filters and column masking for fine-grained access controls.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 58 Protect data with filters and masking

RISKS	DESCRIPTION	CONTROL CATEGORY
MODEL SERVING — INFERENCE REQUESTS 9.10 MODEL SERVING — INFERENCE REQUESTS 9.13 MODEL SERVING - INFERENCE RESPONSE 10.6	Implement filters on sensitive table data using row filters and column masks by harnessing the power of Unity Catalog to secure your data at a granular level. Row filters allow you to apply a filter to a table so that queries return only rows that meet the filter criteria. Column masks let you apply a masking function to a table column.	Implementation PRODUCT REFERENCE AWS | Azure | GCP

DASF 59 Use clean rooms

RISKS	DESCRIPTION	CONTROL CATEGORY
RAW DATA 1.1 RAW DATA 1.7 MODEL MANAGEMENT 8.2 MODEL SERVING — INFERENCE REQUESTS 9.10	Building AI applications today necessitates collaborative efforts across organizations and teams, emphasizing a commitment to privacy and data security. Databricks Clean Rooms offers a secure environment for private collaboration on diverse data and AI tasks, spanning machine learning, SQL queries, Python, R and more. Designed to facilitate seamless collaboration across different cloud and data platforms, Databricks Clean Rooms ensures multiparty collaboration without compromising data privacy or security and enables organizations to build scalable AI applications in a privacy-safe manner.	Implementation PRODUCT REFERENCE AWS | Azure | GCP

DASF 60 Rate limit number of inference queries

RISKS	DESCRIPTION	CONTROL CATEGORY
MODEL SERVING — INFERENCE REQUESTS 9.1 MODEL SERVING — INFERENCE REQUESTS 9.2 MODEL SERVING — INFERENCE REQUESTS 9.5 MODEL SERVING — INFERENCE REQUESTS 9.6 MODEL SERVING — INFERENCE REQUESTS 9.7 MODEL SERVING - INFERENCE RESPONSE 10.2 MODEL SERVING - INFERENCE RESPONSE 10.3 MODEL SERVING - INFERENCE RESPONSE 10.4 MODEL SERVING - INFERENCE RESPONSE 10.5 MODEL SERVING - INFERENCE RESPONSE 10.6	Enforce request-rate limits to manage traffic at the endpoint level on a per-user and per-endpoint basis, effectively controlling access levels and volume.	Configuration PRODUCT REFERENCE AWS | Azure | GCP

DASF 61 Train users on AI/ML security

RISKS	DESCRIPTION	CONTROL CATEGORY
	Provide secure coding education and AI vulnerability awareness for model developers. Training personnel who manage AI infrastructure on cybersecurity best practices is essential to prevent human errors that could lead to security breaches.	Academy PRODUCT REFERENCE See Resources and Further Reading

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM	

DASF 62 Implement network segmentation

RISKS	DESCRIPTION	CONTROL CATEGORY
MODEL SERVING — INFERENCE REQUESTS 9.13	Establish network and security policies to define and enforce egress rules from models and outbound network connections from your serverless compute resources.	Configuration PRODUCT REFERENCE AWS | Azure | N/A

DASF 63 Update software

RISKS	DESCRIPTION	CONTROL CATEGORY
PLATFORM 12.1	Patch and update software regularly to address existing and potential vulnerabilities throughout the AI lifecycle. In Databricks, automatic cluster updates ensure that all clusters within a workspace receive the latest OS images and security patches periodically. Account administrators can customize the maintenance window frequency, start date and time.	Out-of-the-box PRODUCT REFERENCE AWS | Azure | N/A

DASF 64 Limit access from AI models and agents

RISKS	DESCRIPTION	CONTROL CATEGORY
MODEL SERVING — INFERENCE REQUESTS 9.12 MODEL SERVING — INFERENCE REQUESTS 9.13	Allow AI models and agents access to enterprise resources based on the principle of least privilege. In Unity Catalog, a “securable object” is any object that can be permissioned to a principal (e.g., user, service principal or group) and is organized hierarchically. Treat AI as a principal and assign permissions accordingly.	Configuration PRODUCT REFERENCE AWS | Azure | GCP

04 Conclusion

In an era defined by data-driven decision-making and intelligent automation, the importance of AI security and risk management can't be overstated. The Databricks AI Security Framework provides essential guidance for securely developing, deploying and maintaining AI models at scale — and assisting organizations in ensuring their AI models remain secure and continue to deliver business value. The emergence of AI highlights the rapid advancement and specialized needs of its security. However, at its heart, AI security is still rooted in the foundational principles of cybersecurity. Data, AI and security teams must actively collaborate to pursue their common goal of improving the security of AI systems while delivering value.

We've developed a companion compendium document (Google sheet, Excel) for the Databricks AI Security Framework (DASF), which is available for download. This compendium serves as a versatile tool for practitioners, enabling them to engage with DASF by organizing and applying its risks, threats, controls and mappings to industry-recognized standards like MITRE, OWASP, NIST, ISO, HITRUST and more. It allows practitioners to tailor DASF guidelines to meet their organization's AI security needs. Through this document, teams can:

- Assess and categorize risks across various AI system components, from raw data to machine learning models
- Map appropriate security controls to identified risks, facilitating targeted mitigation efforts
- Adapt controls to address specific requirements, such as regulatory compliance, internal audits or governance standards
- Track security measure implementation, identifying any gaps that need further attention
- Map risks and controls to widely recognized standards, including MITRE ATLAS and ATT&CK, OWASP's LLM and ML top 10, NIST, ISO and HITRUST

Please use the companion compendium, watch the DASF walkthrough video on the Databricks Security Best Practices YouTube channel and follow the steps below to get guidance for implementation controls. Whether you're implementing traditional machine learning solutions or LLM-driven applications, the core tenets of AI adoption remain constant.

Databricks AI Security Framework (DASF)

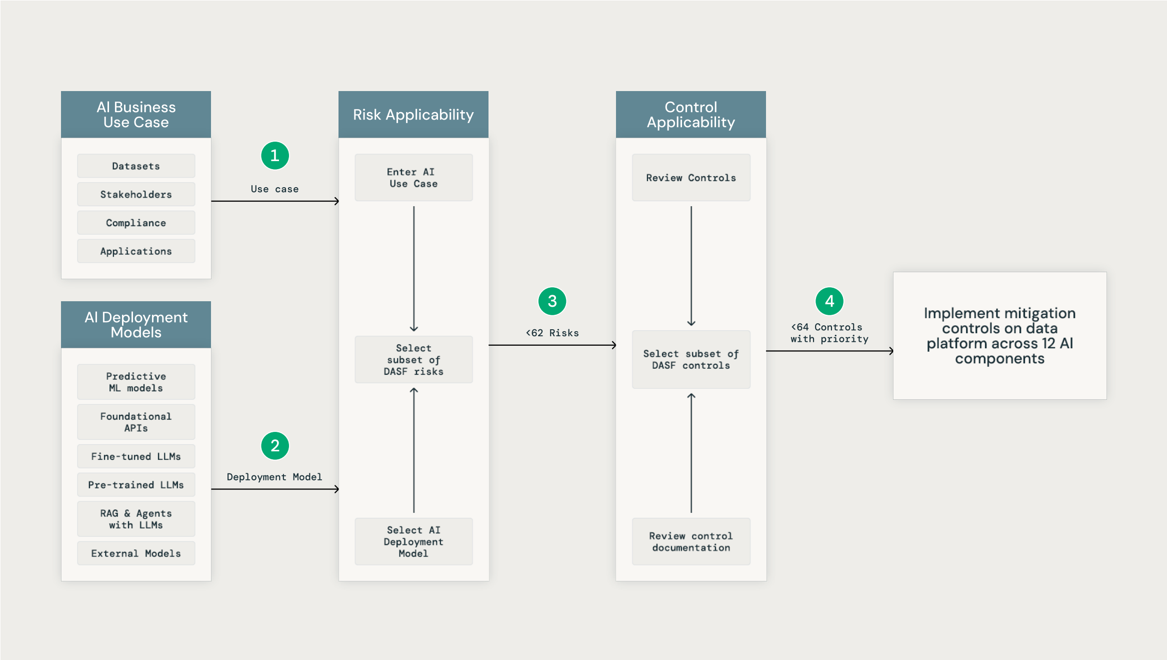


Figure 2: Implementation guidance of DASF controls on the Databricks Data Intelligence Platform

1. **Identify the AI business use case:** Always keep your business goals in mind. Make sure there's a well-defined use case with stakeholders you're trying to secure adequately, whether already implemented or in planning phases. Note the name and type of use case. Example use cases are Generative AI Use Case, ML: Computer Vision Use Case, ML: Natural Language Processing Use Case, ML: Recommendation System Use Case, ML: Predictive Analytics Use Case, ML: Decisioning Use Case.

2. **Determine the AI deployment model:** Choose an appropriate deployment model (e.g., predictive ML models, Foundation Model APIs, RAG LLMs, fine-tuned LLMs and pretrained LLMs, as described in Section: 1.2 How to use this document). Review the "AI Lifecycle Risks" worksheet in the compendium and familiarize yourself with the risks in 12 AI system components, mappings to Deployment models column, Mitigation Controls for each risk and details of the risks in the worksheet "Databricks AI Mitigation Controls." Make a note of the deployment model that is applicable to your use case.

3. **Analyze DASF risk and control applicability:** Using the compendium worksheet
"DASF Risk Applicability" input:

   "AI Deployment Model" value that you noted in step 2

   "AI Use Case" that you noted in step 1

   "Datasets" that you're using with this use case

   "Stakeholders" for the use case. This will produce the list of DASF controls in the worksheet "DASF Control Applicability."

4. **Review and implement mitigation controls:** Review the list of mitigation controls in the worksheet "DASF Control Applicability" that align with your organization's risk appetite. These controls are defined generically for compatibility with any data platform. Our framework also provides guidelines on tailoring these controls specifically for the Databricks Data Intelligence Platform with specific Databricks product references by cloud. You use these controls alongside your organization's policies and have the right assurance in place.

While manually checking DASF controls is important when first configuring Databricks, we've produced the Security Analysis Tool (SAT) to help you monitor the security health of your Databricks Platform. We recommend that you set up SAT against all workspaces so that you can review your deployment configurations against our best practices on a continuous basis. Learn more!

As we embrace the ongoing wave of AI advancements, it's clear that employing a robust, secure MLOps strategy will remain central to unlocking AI's full potential. With firm, secure MLOps foundations in place, organizations will be able to maximize their AI investments to drive innovation and deliver business value. Red teaming and testing can help iteratively improve and mitigate discovered weaknesses of models.

A lot of care has been taken to be accurate. However, as AI is an evolving field, we're happy to have feedback. Please reach out to us if you have any feedback. If you're interested in participating in one of our AI risk workshops, please contact dasf@databricks.com.

The Databricks Data Intelligence Platform stands uniquely positioned as a secure, unified, data-centric platform for both MLOps and LLMOps by taking a defense-in-depth approach to helping organizations implement security across all AI system components. Learn more about Databricks in the Appendix. If you're curious about how Databricks approaches security, please visit our Security and Trust Center.

Resources and Further Reading

We've discussed many different capabilities in this document, with documentation links where possible. Organizations that prioritize high security can learn more than what's in this document. Here are additional resources to dive deeper.

AI and machine learning on Databricks (Mosaic AI)

Training course: Generative AI Fundamentals →

Web page: AI and Machine Learning on Databricks →

Industry solutions: Solution Accelerators →

Blogs: AI/ML Blogs →

eBooks: Big Book of Generative AI → | Big Book of MLOps: 2nd Edition → | Data, Analytics and AI Governance →

Learning library: Generative AI Engineering With Databricks →

DASF video: Introducing the Databricks AI Security Framework (DASF) to Manage AI Security Risks →

Databricks Unity Catalog

Web pages: Databricks Unity Catalog → | AI Governance →

eBook: Data and AI Governance →

Blogs: What's New in Unity Catalog → | Open Sourcing Unity Catalog → | Row and Column Level Security →

Databricks Platform security

Review the security features in the Security and Trust Center, along with the overall documentation about the Databricks security and compliance programs.

The Security and Trust Overview Whitepaper provides an outline of the Databricks architecture and platform security practices.

Databricks Platform Security Best Practices | AWS | Azure | GCP

Security Analysis Tool (SAT)

Databricks Security and Trust Blog

Databricks Security Best Practices YouTube channel

Databricks Delta UniForm

Web page: Delta Lake UniForm →

eBook: O'Reilly Delta Lake: The Definitive Guide →

Blog: General Availability of Delta Lake UniForm →

Responsible AI on the Databricks Data Intelligence Platform

Web page: Responsible AI →

Blogs: Responsible AI with the Databricks Data Intelligence Platform →
Helping Enterprises Responsibly Deploy AI →
Partnerships With Industry and Government Organizations →
AI Regulations and the Databricks Data Intelligence Platform →

Data sharing and collaboration

Web page: Delta Sharing → | Databricks Clean Rooms →

eBooks: Data Sharing and Collaboration With Delta Sharing → | The Definitive Guide to Data Clean Rooms →

Blogs: What's New in Data Sharing and Collaboration → | AI Model Sharing → | How Delta Sharing Enables Secure End-to-End Collaboration → | Clean Rooms Public Preview →

Industry resources

An Architectural Risk Analysis of Machine Learning Systems →

NIST AI Risk Management Framework →

MITRE ATLAS Adversarial ML →

MITRE ATT&CK →

OWASP Top 10 for ML →

OWASP Top 10 for LLMs →

ISO/IEC 42001:2023 →

ISO 27001:2022 →

ISO/IEC 38507:2022 →

NIST – SP 800-53 – Rev 5 →

Guidelines for Secure AI System Development →

Generative AI Framework for HMG →

NIST Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations →

Secure by Design — Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software →

Multilayer Framework for Good Cybersecurity Practices for AI →

HITRUST AI Security Certification Specification (draft) →

SAFE Security AI Risk Management →

The information in this document does not constitute or imply endorsement or recommendation of any third-party organization, product or service by Databricks. Links and references to websites and third-party materials are provided for informational purposes only and do not represent endorsement or recommendation of such resources over others.

Acknowledgments

This whitepaper would not be possible without the insight and guidance provided by our reviewers and contributors at Databricks and externally. Additionally, we extend our appreciation to the frameworks that inspired our research (MITRE, OWASP, NIST, BIML, etc.), as they have played a pivotal role in shaping the foundation of the Databricks AI Security Framework.

We would like to thank the following reviewers and contributors:

DATABRICKS

Matei Zaharia
Chief Technology Officer and Co-Founder

Fermín Serna
Chief Security Office

Omar Khawaja
Vice President, Field CISO

Arun Pamulapati
Senior Staff Security Field Engineer

David Wells
Staff Security Field Engineer

Kelly Albano
Product Marketing Manager

Abhi Arikapudi
Senior Director Security Engineering

David Veuve
Head of Security Field Engineering

Tim Lortz
Lead Specialist Solutions Architect

Joseph Bradley
Principal ML Product Specialist

Arthur Dooner
Specialist Solutions Architect

Veronica Gomes
Solutions Architect

Aliaksandra Nita
Senior Technical Program Manager

Neil Archibald
Senior Staff Security Engineer

Silvio Fiorito
Principal Security Field Engineer

Andrew Weaver
Principal Specialist Solutions Architect

Tony Bo
Sr. Specialist Solutions Architect

Muhammad Aun
Sr. Incident Handler

Keana Robles
Technical Program Manager

Jesse Scott
Sr. Industry Solutions Director, Cybersecurity Go-To-Market

Suchismita Pahi
Lead Counsel, Product

ROBUST INTELLIGENCE

Hyrum Anderson
Chief Technology Officer

Alie Fordyce
Product Policy

Adam Swanda
AI Security Researcher – Threat Intelligence

NAVY FEDERAL CREDIT UNION

Riyaz Poonawala
Vice President, Information Security

PROTECT AI

Diana Kelley
CISO

BARRACUDA

Grizel Lopez
Sr. Director of Engineering

CARNEGIE MELLON UNIVERSITY

Hasan Yasar
Technical Director, Teaching Professor, Continuous Deployment of Capability Software Engineering Institute

META

Brandon Sloane
Risk Lead

CAPITAL ONE FINANCIAL

Ebrima N. Ceesay
PhD, CISSP, Senior Distinguished Engineer

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

HIDDENLAYER

Christopher Sestito
Co-founder & CEO

Abigail Maines
CRO

Hiep Dang
VP of Strategic Tech Alliances

HITRUST

Robert Booker
EVP Strategy, Research and Innovation Center of Excellence and Chief Strategy Officer

Jeremy Huval,
Chief Innovation Officer

COMPLYLEFT

Ben Johns
Cybersecurity Specialist

U.S. DEPARTMENT OF VETERAN AFFAIRS

Joseph Raetano
Artificial Intelligence (AI) Lead, Summit Data Analytics & AI Platform (SDP)

ETHRIVA

Nisar Khan
AI Security Researcher

Kal Chakravarthi
Founder

Ananya Gangavarapu
AI Engineer

ARHASI

Chiru Bhavansikar
Chief AI Officer

Melody Roth
Exec. Director, AI Strategy & Operations

Shayaan Hussain
Lead GenAI Engineer

Nimisha Bhide
Lead Data Engineer

THE FAIR INSTITUTE & SAFE SECURITY

Jacqueline Lebo
Risk Advisory Manager

KYTHERA LABS & SUNNYDATA.AI

Josue A. Bogran
Architect & Advisor

ZILLOW GROUP, INC.

Ben Vardag
Manager, Data Governance

STATE STREET CORPORATION

Ajish D. George
PhD, Managing Director, Cybersecurity Architecture and Fusion Engineering

BNP PARIBAS

Sandip Wadje
Managing Director - Global Head of Emerging Technology Risks

BARCLAYS PLC

Yaman Saqqa
Head of Security Product Engineering

THE ING GROUP

Eduardo Barbaro
PhD, Head of Security Analytics

IBM

Purushotam Shrestha
Data Engineer and Databricks SME

THE COLLEGE BOARD

Shilpa Jasthi
Head of Information Security, Infrastructure and Workplace Technology

JPMORGAN CHASE

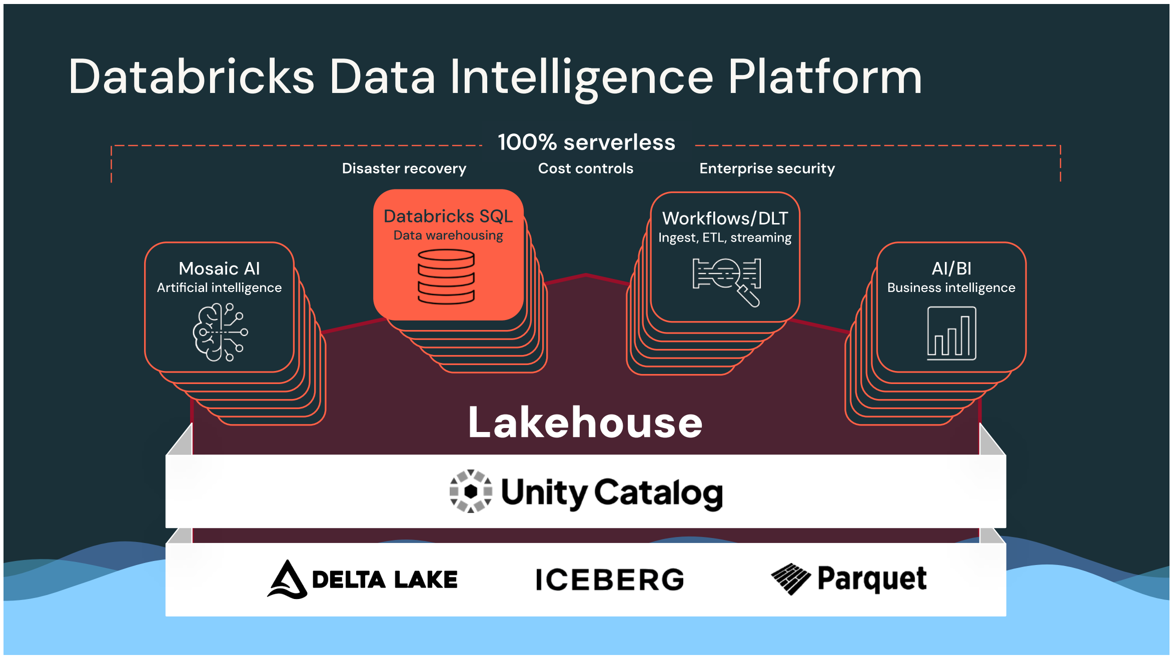
Rakesh Patil
Director of Cybersecurity Architecture , AI/ML Data platforms

07 Appendix: Understanding the Databricks Data Intelligence Platform

Databricks is the data and AI company with origins in academia and the open source community. Databricks was founded in 2013 by the original creators of Apache Spark™, Delta Lake and MLflow. We pioneered the concept of the lakehouse to combine and unify the best of data warehouses and data lakes. Databricks made this vision a reality in 2020; since then, it has seen tremendous adoption as a category. Today, 74% of global CIOs report having a lakehouse in their estate, and almost all of the remainder intend to have one within the next three years.

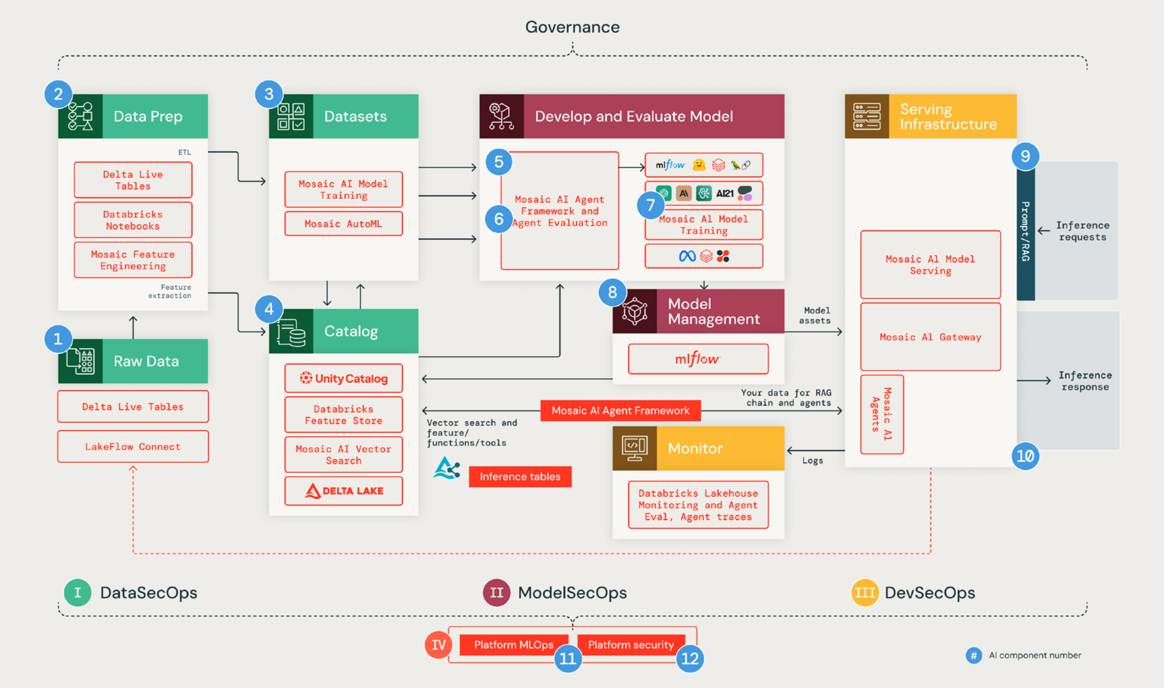
In November 2023, we announced the Databricks Data Intelligence Platform. It's built on a lakehouse to provide an open, unified foundation for all data and governance. We built the Data Intelligence Platform to allow every employee in every organization to find success with data and AI. At the heart of the platform is a Data Intelligence Engine, DatabricksIQ, that understands the semantics of your data and how it flows across all of your workloads. This allows for new methods of optimization, as well as for technical and nontechnical users to use natural language to discover and use data and AI in the context of your business.

In this section, we provide an overview of our platform and its architecture and components related to governance, security, and AI and machine learning.



The Databricks Data Intelligence Platform combines AI assets — from data and features to models — into one catalog, ensuring full visibility and fine-grained control throughout the AI workflow. We provide automatic lineage tracking, centralized governance and seamless cross-workspace collaboration for simplified MLOps and enhanced productivity. Furthermore, we give customers complete control and ownership of their data and models with privacy controls to maintain compliance as well as efficiency and granular models on their data, fine-tuned at lower costs.

The Databricks Data Intelligence Platform offers a secure, unified and data-centric solution for both MLOps and LLMOps, adopting a defense-in-depth approach to implementing security across all AI system components. As shown in the following diagram, the platform aligns seamlessly with the AI system components defined in the DASF, supporting key stages — from data preparation to serving infrastructure — through products like Delta Live Tables, Unity Catalog and Mosaic AI.



Databricks Platform architecture

Databricks is a platform as a service (PaaS) general-purpose data-agnostic compute platform.

We use the phrase “hybrid PaaS” because our lakehouse architecture is split into two separate planes to simplify your permissions, avoid data duplication and reduce risk. The control plane is the management plane where Databricks runs the workspace application and manages notebooks, configuration and clusters. The compute plane handles your data processing. Customers deploy a compute plane (virtual network and compute) in a cloud service provider account (such as AWS, Azure or GCP) that the customer owns. With serverless deployments, the compute plane exists in the customer’s Databricks account rather than their cloud service provider account. Customers get the benefits of PaaS with the option to keep their data processing clusters locally within their environment.

The phrase “general-purpose data-agnostic” means that you can use Databricks services for any type of data and purpose that you need. If you’re new to Databricks or the lakehouse architecture, start with an overview of the architecture and a review of common security questions before you hop into specific recommendations. You’ll see those in our Security and Trust Center and the Security and Trust Overview Whitepaper.

Controls addressed by Databricks Platform architecture:

- DASF 1: SSO with IdP and MFA
- DASF 2: Sync users and groups
- DASF 3: Restrict access using IP access lists
- DASF 4: Restrict access using private link
- DASF 5: Control access to data and other objects
- DASF 56: Restrict outbound connections from models
- DASF 62: Implement network segmentation
- DASF 64: Limit access from AI models and agents

Delta Lake

Delta Lake is the optimized storage layer that provides the foundation for tables in a lakehouse on Databricks. Delta Lake is open source software that extends Parquet data files with a file-based transaction log for ACID transactions and scalable metadata handling. It's fully compatible with Apache Spark APIs and was developed for tight integration with Structured Streaming, allowing you to easily use a single copy of data for both batch and streaming operations and providing incremental processing at scale.

Universal Format (UniForm), is a feature of Delta Lake that takes advantage of the inherent similarities among the three open table formats. Delta Lake, Iceberg and Apache Hudi all store data in the Apache Parquet file format but diverge in how they store additional metadata. UniForm generates Iceberg metadata alongside Delta Lake while maintaining a single copy of the Parquet files. By writing once to Delta Lake, you can access your data using any engine that supports any one of the open formats. Additional details:

- Delta Lake is the default format for all operations on Databricks. Unless otherwise specified, all tables on Databricks are Delta tables.
- Delta Lake is deeply integrated with Spark Structured Streaming through readStream and writeStream. Delta Lake overcomes many of the limitations typically associated with streaming systems and files.
- Additionally, you can use Auto Loader to incrementally and efficiently process new data files as they arrive in cloud storage. It provides a Structured Streaming source called cloudFiles. Given an input directory path on the cloud file storage, the cloudFiles source automatically processes new files as they arrive, with the option of also processing existing files in that directory. Auto Loader has support for both Python and SQL in Delta Live Tables. Auto Loader can process billions of files to migrate or backfill a table. Auto Loader scales to support near real-time ingestion of millions of files per hour.
- Liquid clustering replaces table partitioning and Z-Ordering to simplify data layout decisions and optimize query performance. Liquid clustering provides flexibility to redefine clustering keys without rewriting existing data, allowing data layout to evolve alongside analytic needs over time. Databricks recommends liquid clustering for all new Delta tables.
- Predictive optimization automatically optimizes your data for the best performance and price. It learns from your data usage patterns, builds a plan for the right optimizations to perform and then runs those optimizations on hyper-optimized serverless infrastructure.

Controls addressed by Delta Lake:

- DASF 7: Enforce data quality checks on batch and streaming datasets
- DASF 10: Version data
- DASF 12: Delete records from datasets
- DASF 13: Use near real-time data

Databricks Unity Catalog

Databricks Unity Catalog is the industry's only unified and open governance solution for managing data and AI assets across any lakehouse format or data source. Unity Catalog's security model is based on standard ANSI SQL and allows administrators to grant permissions in their existing data lake using familiar syntax at the level of catalogs, schemas (also called databases), tables and views. With Unity Catalog, data scientists, analysts and engineers can seamlessly govern their structured and unstructured data, machine learning models, notebooks, dashboards and files on any cloud or platform. This unified approach to governance accelerates data and AI initiatives while ensuring regulatory compliance in a simplified manner.

Unity Catalog provides key capabilities like:

- **Access control for data and AI assets with a single permission model:** Unity Catalog simplifies access management with a unified interface to define access policies on data and AI assets and consistently apply and audit these policies on any cloud or data platform. You can access data from other computing platforms using open interfaces, with consistent permissions managed in one place.
- **Open data sharing and collaboration:** Easily share data and AI assets across clouds, regions and platforms with open source Delta Sharing, natively integrated within Unity Catalog. Securely collaborate with anyone, anywhere to unlock new revenue streams and drive business value without relying on proprietary formats, complex ETL processes or costly data replication.
- **Centralized data search and discovery:** Quickly find, understand and reference data from across your data estate, boosting productivity. Data search in Unity Catalog is secure by default, limiting search results based on access privileges of the users and adding an additional layer of security for privacy considerations.
  - Tags are attributes that include keys and optional values that you can use to organize and categorize securable objects in Unity Catalog. They simplify search and discovery of tables and views.

- **Automated lineage:** You can use Unity Catalog to capture runtime data lineage across queries in any language executed on a Databricks cluster or SQL warehouse. Lineage is captured down to the column level, and includes notebooks, jobs and dashboards related to the query. Lineage can be retrieved via REST APIs to support integrations with our catalog partners.
- **Historical observability across your account with system tables:** System tables are a Databricks-hosted analytical store of your account's operational data found in the system catalog. Unity Catalog lets you easily access and query your account's operational data, including audit logs, billable usage and lineage.
- **Built-in auditing:** Unity Catalog automatically captures user-level audit logs that record access to your data
- **Row filters and column masking:** This capability allows you to secure and govern your data at the granular level.
  - Row filters allow you to apply a filter to a table. You implement a row filter as a SQL user-defined function (UDF). Python and Scala UDFs are also supported, but only when they are wrapped in SQL UDFs.
  - Column masks let you apply a masking function to a table column. The masking function evaluates at query runtime, substituting each reference of the target column with the results of the masking function. For most use cases, column masks determine whether to return the original column value or redact it based on the identity of the invoking user.
- **Attribute-based access controls (ABAC) — in Private Preview:** ABAC offers organizations a high-leverage governance solution that simplifies the enforcement of governance policies across their entire lakehouse. By employing straightforward rules and tags, ABAC ensures consistent governance across all data sources, whether native to Databricks or federated from external sources. With ABAC, users can establish access controls tailored to specific attributes of resources like workspaces, data assets such as tables, and AI assets. These attributes encompass a wide range of parameters, including user-defined tags, workspace details, location, identity and time. Whether it's ensuring sensitive data remains restricted to authorized personnel or dynamically adjusting access based on changing project requirements, ABAC empowers users to enforce security measures with granular precision.

- **Model management:** Models in Unity Catalog extends the benefits of Unity Catalog to ML models, including centralized access control, auditing, lineage and model discovery across workspaces
- **Lakehouse monitoring:** Databricks Lakehouse Monitoring lets you monitor the statistical properties and quality of the data in all of the tables in your account. You can also use it to track the performance of machine learning models and model-serving endpoints by monitoring inference tables that contain model inputs and predictions.
- **Lakehouse Federation:** Lakehouse Federation is the query federation platform for Databricks. The term “query federation” describes a collection of features that enable users and systems to run queries against multiple data sources without needing to migrate all data to a unified system. Databricks uses Unity Catalog to manage query federation. You configure read-only connections to popular database solutions using drivers that are included on pro SQL warehouses, serverless SQL warehouses and Databricks Runtime clusters.

Controls addressed by Databricks Unity Catalog:

- DASF 2: Sync users and groups
- DASF 6: Classify data
- DASF 11: Capture and view data lineage
- DASF 12: Delete records from datasets
- DASF 14: Audit actions performed on datasets
- DASF 16: Secure model features
- DASF 17: Track and reproduce the training data used for ML model training
- DASF 18: Govern model assets
- DASF 21: Monitor data and AI system from a single pane of glass
- DASF 23: Register, version, approve, promote, deploy models
- DASF 24: Control access to models and model assets
- DASF 25: Use retrieval augmented generation (RAG) with large language models (LLMs)
- DASF 28: Create model aliases, tags and annotations
- DASF 30: Encrypt models
- DASF 32: Streamline the usage and management of various large language model (LLM) providers

- DASF 35: Track model performance
- DASF 37: Set up inference tables for monitoring and debugging models
- DASF 46: Store and retrieve embeddings securely
- DASF 47: Compare LLM outputs on set prompts
- DASF 51: Share data and AI assets securely
- DASF 53: Third-party library control
- DASF 54: Implement AI guardrails
- DASF 55: Monitor audit logs
- DASF 57: Use attribute-based access controls (ABAC)
- DASF 58: Protect data with filters and masking

Databricks Platform security

Data and AI are your most valuable assets and always have to be protected — that's why security is built into every layer of the Databricks Data Intelligence Platform. Databricks security is based on three core principles: trust, technology and transparency.

- **Trust** — Third-party audit firms regularly audit Databricks systems and processes. Databricks customers can trust independent validation of internal security processes.
- **Technology** — Databricks deploys modern technology solutions combined with secure processes across the enterprise to maximize security. Security design and tools are applied throughout. Databricks considers security in the platform architecture design, network security processes, automated penetration testing on the production systems and vulnerability scanning tools during development.
- **Transparency** — Databricks provides customers with full attestation reports (for example, SOC 2 Type 2), certifications (for example, ISO 27001) and detailed architecture overviews. Our transparency enables you to meet your regulatory needs while taking advantage of our platform.

Our Databricks Security team regularly works with customers to securely deploy AI systems on our platform with the appropriate security and governance features. We understand how ML systems are designed for security, teasing out possible security engineering risks and making such risks explicit. Databricks is committed to providing a data intelligence platform where business stakeholders, data engineers, data scientists, ML engineers, data governance officers and data analysts can trust that their data and AI models are secure.

Controls addressed by Databricks Platform security:

- DASF 1: SSO with IdP and MFA
- DASF 2: Sync users and groups
- DASF 3: Restrict access using IP access lists
- DASF 4: Restrict access using private link
- DASF 5: Control access to data and other objects
- DASF 8: Encrypt data at rest
- DASF 9: Encrypt data in transit
- DASF 31: Secure model serving endpoints
- DASF 33: Manage credentials securely
- DASF 34: Run models in multiple layers of isolation
- DASF 36: Set up monitoring alerts
- DASF 38: Platform security — penetration testing, red teaming, bug bounty and vulnerability management
- DASF 39: Platform security — Incident Response Team
- DASF 40: Platform security — internal access
- DASF 41: Platform security — secure SDLC
- DASF 43: Use access control lists
- DASF 46: Store and retrieve embeddings securely
- DASF 48: Use hardened Runtime for Machine Learning
- DASF 50: Platform compliance
- DASF 51: Share data and AI assets securely
- DASF 52: Source code control
- DASF 53: Third-party library control
- DASF 54: Implement AI guardrails
- DASF 55: Monitor audit logs
- DASF 56: Restrict outbound connections from models
- DASF 58: Protect data with filters and masking

- DASF 60: Rate limit number of inference queries
- DASF 61: Train users on AI/ML security
- DASF 62: Implement network segmentation
- DASF 63: Update software
- DASF 64: Limit access from AI models and agents

Executive Summary
Introduction
Risks in AI System Components
Understanding Databricks Risk AI Mitigation Controls
Conclusion
Resources and Further Reading
Acknowledgments
Appendix: Understanding the Databricks Data Intelligence Platform
Appendix: Glossary
License

Serverless egress control

Serverless egress control (SEG) is a security feature specific to serverless compute on Databricks that allows an administrator to implement networking policies that constrain access to:

- Endpoints on the internet such as Google Drive, Box, OpenAI, etc.
- Unauthorized storage resources (S3, ADLS, GCS) on the cloud platform
- Unauthorized on-premises database resources

SEG offers data exfiltration controls natively within Databricks, and admins can enforce a "deny by default" policy, restricting outbound connections only to approved locations such as specific cloud storage destinations. For added flexibility, admins can customize policies per workspace. Additionally, "log-only mode" logs policy violations without blocking connections, enabling you to test security settings confidently. It can be used in conjunction with technologies such as Private Link. For example, an administrator can block access to endpoints on the internet with SEG and then enable private access to specific backend endpoints over Private Link.

Controls addressed by serverless egress control:

- DASF 3: Restrict access using IP access lists
- DASF 31: Secure model serving endpoints
- DASF 43: Use access control lists
- DASF 54: Implement AI guardrails
- DASF 56: Restrict outbound connections from models
- DASF 62: Implement network segmentation
- DASF 64: Limit access from AI models and agents

Databricks Mosaic AI

Databricks provides a scalable, collaborative platform that empowers ML teams to prepare and process data, streamline cross-team collaboration and standardize the full ML lifecycle from experimentation to production, including generative AI and large language models (LLMs). You can build models from scratch and tune existing models on your data. However, it's not just about building and serving models. Databricks Mosaic AI covers the end-to-end AI workflow to help you deploy and manage models all the way through production. Some of our AI offerings include:

- End-to-end retrieval augmented generation (RAG) to build high-quality conversational agents on your data, leveraging Mosaic AI Vector Search for increased relevance and accuracy
- Integration with data-centric applications with leading AI APIs like OpenAI
- Training of predictive ML models either from scratch on an organization's tabular data or by fine-tuning existing models such as MPT and Meta Llama 3.1 to further enhance AI applications with a deep understanding of a target domain
- Efficient and secure serverless inference on your enterprise data and connection to Unity Catalog governance and quality monitoring functionality
- End-to-end MLOps based on the popular MLflow open source project, with all produced data automatically actionable, tracked and monitorable in the lakehouse
- Improved visibility and proactive detection of anomalies in your entire data and AI workflow, reducing risks, time to value, and high operational costs with Databricks Lakehouse Monitoring
- Mosaic AI Agent Framework gives developers the ability to build and deploy high-quality agentic applications with a set of tools on Databricks designed to help build, deploy and evaluate production-quality AI agents like retrieval augmented generation (RAG) applications

We'll now outline specific products and components of Mosaic AI as they relate to the controls outlined in the Databricks AI Security Framework.

Compound AI system

Compound AI systems involve integrating multiple interacting components such as models, retrievers, functions or external tools to perform complex tasks collaboratively. These systems are designed to handle complex tasks that single AI models can accomplish. Unlike standalone models, which focus on narrow tasks like language generation or image classification, compound AI systems use multiple models and other components that can dynamically collaborate to improve performance, decision-making and adaptability. This approach allows more flexibility and control over the system's behavior, making it a preferred architecture for more complex AI applications. For more information, refer to The Shift from Models to Compound AI Systems.

AI agents

While the industry is refining the definition of AI agents, generally, an AI agent is an application capable of making decisions based on data, learning from experience and adapting to new situations over time. Unlike traditional rule-based systems like robotic process automation (RPA), AI agents can manage structured, unstructured and other data types, analyze them and make informed decisions in dynamic or uncertain environments. These agents often use large language models (LLMs) to accomplish their objectives, helping automate tasks and streamline workflows.

AI agents can be classified into two main types:

1. **Interactive agents:** These agents respond directly to human input. A common example is a generative AI (GenAI) chatbot that interacts with users in real time.
2. **Autonomous agents:** These agents operate independently, automating tasks or workflows without human input. They trigger actions in response to events or processes and make decisions based on predefined logic. For instance, an autonomous agent could prioritize incidents in a system like ServiceNow, leveraging past knowledge of incident management.

While LLMs and retrieval augmented generation (RAG) are effective at understanding and generating text, they often face limitations regarding real-world task execution. Real-world scenarios demand linguistic comprehension, dynamic decision-making, task execution and adaptability. AI agents address these gaps by dynamically constructing and executing tasks, interacting with internal and external systems and adapting to changing conditions.

For more information, see From LLMs to AI agents.

MLflow

ML lifecycle management in Databricks is provided by managed MLflow. Databricks provides a fully managed and hosted version of MLflow integrated with enterprise security features, high availability and other Databricks workspace features such as experiment and run management and notebook revision capture. MLflow is an open source platform for managing the end-to-end machine learning lifecycle. MLflow supports Java, Python, R and REST APIs. It has the following primary components:

- **Tracking:** Allows you to track experiments to record and compare parameters and results
- **Models:** Allow you to manage and deploy models from a variety of ML libraries to a variety of model serving and inference platforms
- **Projects:** Allow you to package ML code in a reusable, reproducible form to share with other data scientists or transfer to production
- **Model Registry:** Allows you to centralize a model store for managing full lifecycle stage transitions for models, from staging to production, with capabilities for versioning and annotating. Databricks provides a managed version of the Model Registry in Unity Catalog.
- **Model Serving:** Allows you to host MLflow models as REST endpoints. Databricks provides a unified interface to deploy, govern and query your served AI models.
- **MLflow Tracing for agents:** Using MLflow Tracing you can log, analyze and compare traces across different versions of generative AI applications. It allows you to debug your generative AI Python code and keep track of inputs and responses. Doing so can help you discover conditions or parameters that contribute to poor performance of your application. MLflow Tracing is tightly integrated with Databricks tools and infrastructure, allowing you to store and display all your traces in Databricks Notebooks or the MLflow experiment UI as you run your code.

Controls addressed by MLflow:

- DASF 6: Classify data
- DASF 13: Use near real-time data
- DASF 15: Explore datasets and identify problems
- DASF 17: Track and reproduce the training data used for ML model training
- DASF 18: Govern model assets
- DASF 19: Manage end-to-end machine learning lifecycle
- DASF 20: Track ML training runs
- DASF 21: Monitor data and AI system from a single pane of glass
- DASF 23: Register, version, approve, promote, deploy and monitor models
- DASF 24: Control access to models and model assets
- DASF 26: Fine-tune large language models (LLMs)
- DASF 27: Pretrain a large language model (LLM)
- DASF 28: Create model aliases, tags and annotations
- DASF 29: Build MLOps workflows
- DASF 30: Encrypt models
- DASF 31: Secure model serving endpoints
- DASF 32: Streamline the usage and management of various large language model (LLM) providers
- DASF 35: Track model performance
- DASF 42: Employ data-centric MLOps and LLMOps
- DASF 44: Triggering actions in response to a specific event
- DASF 45: Evaluate models
- DASF 47: Compare LLM outputs on set prompts
- DASF 49: Automate LLM evaluation
- DASF 51: Share data and AI assets securely
- DASF 54: Implement AI guardrails
- DASF 55: Monitor audit logs

Mosaic AI Model Training

With Mosaic AI Model Training (formerly Foundation Model Training), you can use your own data to customize a foundation model to optimize its performance for your specific application. By conducting full parameter fine-tuning or continuing training of a foundation model, you can train your own model using significantly less data, time and compute resources than training a model from scratch.

Controls addressed by Mosaic AI Model Training:

- DASF 22: Build models with all representative, accurate and relevant data sources
- DASF 26: Fine-tune large language models (LLMs)
- DASF 27: Pretrain a large language model (LLM)

Mosaic AI Vector Search

Mosaic AI Vector Search is a vector database that's built into Databricks and integrated with its governance and productivity tools. Mosaic AI Vector Search enables developers to improve the accuracy of their retrieval augmented generation (RAG) and generative AI applications through similarity search over unstructured documents such as PDFs, Microsoft Office documents and wikis. Mosaic AI Vector Search offers the following security features:

- Every customer request to Mosaic AI Vector Search is logically isolated, authenticated and authorized
- Encryption of all data at rest (AES-256) and in transit (TLS 1.2+)
- Integration with Unity Catalog to allow for vector indexes to be stored as entities within Unity Catalog and leveraged under the same unified interface to define policies on data, with fine-grained control on embeddings
- Support for two modes of authentication:
    - Personal access token (PAT): You can use a personal access token to authenticate with Mosaic AI Vector Search. See personal access authentication token. If you use the SDK in a notebook environment, it automatically generates a PAT for authentication.
    - Service principal token: An admin can generate a service principal token and pass it to the SDK or API. See manage service principals. For production use cases, Databricks recommends using a service principal token.
- Customer-managed keys (CMK) are supported on endpoints created on or after May 8, 2024

Controls addressed by Mosaic AI Vector Search:

- DASF 25: Use retrieval augmented generation (RAG) with large language models (LLMs)
- DASF 37: Set up inference tables for monitoring and debugging models
- DASF 46: Store and retrieve embeddings securely

Mosaic AI Gateway

With Mosaic AI Gateway you can streamline the usage and management of generative AI (GenAI) models with your organization. It's a centralized service that brings governance, monitoring and production readiness to model serving endpoints. It also allows you to run, secure and govern AI traffic. Many enterprises mix and match multiple AI models from different providers to build compound AI systems (e.g., RAG, multi-agent architectures) that achieve the quality needed to deploy GenAI applications into production. However, as enterprises integrate a diverse array of open and proprietary models, they encounter challenges with operational inefficiencies, cost overruns and potential security risks. With Mosaic AI Gateway you can configure the following controls on your AI system:

- Permission and rate limiting to control who has access and how much access
- Payload logging to monitor and audit data being sent to model APIs using inference tables
- Usage tracking to monitor operational usage on endpoints and associated costs using system tables
- Traffic routing to minimize production outages during and after deployment

AI guardrails is another control within Mosaic AI Gateway. With AI guardrails you can configure and enforce data compliance at the model serving endpoint level to reduce harmful content on any requests sent to the underlying model. AI guardrails has the following controls:

- Safety filtering prevents your model from interacting with unsafe and harmful content like violent crime, self-harm and hate speech
- Personally identifiable information (PII) detection to detect any sensitive information (such as names, addresses, credit card numbers) for users
- Topic moderation to list a set of allowed topics. Given a chat request, this guardrail flags the request if its topic isn't in the allowed topics.
- Keyword filtering to specify different sets of invalid keywords for both the input and the output. One potential use case for keyword filtering is to prevent the model from talking about competitors.

Controls addressed by Mosaic AI Gateway:

- DASF 11: Capture and view data lineage
- DASF 14: Audit actions performed on datasets
- DASF 32: Streamline the usage and management of various large language model (LLM) providers
- DASF 33: Manage credentials securely
- DASF 43: Use access control lists
- DASF 45: Evaluate models
- DASF 51: Share data and AI assets securely
- DASF 54: Implement AI guardrails
- DASF 55: Monitor audit logs
- DASF 60: Rate limit number of inference queries

Databricks Lakehouse Monitoring

Databricks Lakehouse Monitoring lets you monitor the statistical properties and quality of the data in all of the tables in your account. You can also use it to track the performance of machine learning models and model serving endpoints by monitoring inference tables that contain model inputs and predictions.

To draw useful insights from your data, you must have confidence in the quality of your data. Monitoring your data provides quantitative measures that help you track and confirm the quality and consistency of your data over time. When you detect changes in your table's data distribution or corresponding model's performance, the tables created by Databricks Lakehouse Monitoring can capture and alert you to the change and can help you identify the cause.

Controls addressed by Databricks Lakehouse Monitoring:

- DASF 35: Track model performance
- DASF 55: Monitor audit logs

Databricks Clean Rooms

Databricks Clean Rooms uses Delta Sharing and serverless compute to provide a secure and privacy-protecting environment where multiple parties can work together on sensitive enterprise data without direct access to each other's data. Databricks Clean Rooms enables customers to execute diverse workloads using their preferred languages like SQL, Python and soon, Scala and Java. It supports multicloud collaboration across platforms such as AWS, Azure and GCP, with upcoming support for federated queries with external data platforms like Snowflake and BigQuery. Currently supporting two-party collaboration, Databricks plans to scale up to 10 collaborators post-GA, offering APIs and orchestration workflows for flexibility.

Controls addressed by Databricks Clean Rooms:

- DASF 59: Use clean rooms

Databricks Git folders

Databricks Git folders (formerly known as "Repos") is a visual Git client in Databricks. It supports common Git operations such as cloning a repository, committing and pushing, pulling, branch management and visual comparison of diffs when committing. Within Databricks Git folders you can develop code in notebooks or other files and follow data science and engineering code development best practices using Git for version control, collaboration and CI/CD.

Controls addressed by Databricks Git folders:

- DASF 52: Source code control

Databricks approach to AI red teaming

Databricks believes that the advancement of AI relies on building trust in intelligent applications by following responsible practices in its development and use. This requires that every organization has ownership and control over their data and AI models with comprehensive monitoring, privacy controls and governance throughout the AI development and deployment. AI red teaming, especially for large language models, is essential to developing and deploying models safely. The tactics, tools and procedures used in AI red teaming vary depending on the scope of the engagement and the modality of the models being tested (e.g. large language models, vision models or hybrid systems).

- **Large language models (LLMs):** Testing focuses on prompt manipulation, bias detection and harmful output prevention. Techniques such as adversarial prompt crafting and semantic manipulation are essential.
- **Vision models:** Emphasis is placed on adversarial images, pixel-level attacks and testing robustness to occlusion, noise and transformations
- **Hybrid models:** When models combine multiple modalities, the interaction between different subsystems is carefully examined, ensuring that vulnerabilities in one modality do not cascade into systemwide failures

AI red teaming is an adaptation of the traditional concept of "red teaming," a term which originated in military and intelligence contexts. Traditionally, red teaming refers to the practice of adopting an adversarial role to simulate potential threats or challenges to a system or process in order to identify vulnerabilities and weaknesses.

In the case of AI red teaming, the approach is applied to AI systems, where ethical hackers or AI specialists play the role of adversaries to uncover risks such as biases, security vulnerabilities or unintended behaviors in AI models and their associated applications.

This section outlines our general approach, highlighting the distinction between automated and manual testing methods and the technologies and frameworks employed to maximize the effectiveness of the red teaming process. It isn't a full exploration of our lab but rather a sample subset of the techniques, tools and processes we employ.

Please see the companion compendium document for a collection of third-party tools for "Public AI Red Teaming Tools."

Traditional model AI testing

Traditional model AI testing often involves the creation of adversarial examples, which are specially crafted inputs designed to trick or mislead the model into making incorrect predictions. These adversarial examples exploit vulnerabilities in the model by introducing subtle, often imperceptible changes to input data (such as images, text or audio) that result in dramatically different outputs. In many traditional testing setups, these adversarial examples are used to evaluate the model's robustness and retrain it, helping the AI system learn to recognize and resist such manipulations. This iterative process aims to strengthen the model's defenses against adversarial attacks by refining its ability to handle edge cases and abnormal inputs more reliably.

GenAI red teaming

At Databricks, our approach to GenAI red teaming combines automated testing to map out gaps in model alignment and manual testing to explore the boundaries and edge cases of those gaps. Both methods complement each other in identifying weaknesses and ensuring that AI models meet the expected safety and alignment standards.

AUTOMATED TESTING: IDENTIFYING ALIGNMENT GAPS

Automated testing is the first step in the process, where we systematically leverage tools and technologies to identify potential issues in the model's behavior. This step helps to assess the model's performance across various scenarios quickly.

DATASET ACQUISITION AND AUGMENTATION

We rely on industry-standard and AI-generated data to ensure comprehensive testing to simulate multiple attack scenarios and create a robust and diverse testing environment, maximizing the likelihood of discovering issues with the model.

- **Industry-standard datasets:** We utilize well-established datasets that are widely recognized across the industry for specific modalities. These datasets provide a strong baseline for initial testing, ensuring that the model is evaluated on accepted benchmarks.

  The following are examples of some valuable datasets for evaluating GenAI models:

  - StereoSet: Measuring stereotypical bias in pretrained language models
  - allenai/real-toxicity-prompts · Datasets at Hugging Face
  - GitHub - 0xk1h0/ChatGPT_DAN: ChatGPT DAN, Jailbreaks prompt
  - AIML-TUDA/i2p · Datasets at Hugging Face
- **Dataset expansion via AI augmentation:** We augment standard datasets using AI models. These models can generate new data points or modify existing datasets, introducing slight variations or rare edge cases that help test the model's robustness.
  - **Synthetic data generation** — AI models create entirely new synthetic datasets designed to challenge the model with rare or unanticipated inputs in real-world use
  - **Data augmentation techniques** — We modify existing datasets by introducing changes such as noise, transformations and adversarial inputs. This approach allows us to test how the model adapts to slight but meaningful deviations in input data.

AUTOMATED SCORING

When conducting automated testing, it's important to score the results as well. For this purpose, we employ scoring systems to assess the model's outputs, emphasizing alignment, accuracy and appropriateness whenever possible.

- **Fine-tuned classifiers:** One approach involves using fine-tuned classifiers trained in-house or sourced from platforms like Hugging Face. These classifiers can automatically assess whether the model outputs meet the expected alignment, relevance and correctness criteria.
- **LLMs or MLLMs as judges:** We also use LLMs or multimodal large language models (MLLMs) to evaluate responses. These models serve as automated "judges," providing feedback on the appropriateness, safety and consistency of the model outputs based on predefined criteria.

RESPONSE FORMAT ENFORCEMENT

We leverage software, such as the Rigging module, to enforce strict adherence to the expected format and content of model responses. This ensures that:

- The model's outputs are structured according to a specified format
- Any deviations from the established boundaries are flagged, and the inference can be repeated

AUTOMATION FRAMEWORKS

Automation frameworks such as PyRIT or Garak allow our team to streamline integrating, testing and scoring AI models based on preset performance criteria. The extensive body of previous tests is applied with replay to ensure consistency in reevaluation and facilitating regression testing whenever models are updated. This enables iterative improvements and increasingly robust training cycles. Additionally, automated logging and reporting features within these frameworks support comprehensive documentation.

MANUAL TESTING: BOUNDARY EXPLORATION

Once we've mapped out gaps using automated tools, manual testing allows us to dive deeper into said gaps and explore edge cases where automated systems might miss critical failures. This phase involves human expertise to push models beyond their normal operational limits.

ADVERSARIAL PROMPTS AND EXPLORATORY TESTING

Manual testers focus on probing the boundaries identified in the automated phase. They create adversarial prompts, edge-case scenarios and contextually complex inputs to test how the model handles ambiguous, misleading or highly nuanced situations.

INTEGRATING NEW TECHNIQUES

As AI red teaming practices evolve, we integrate cutting-edge tools and techniques to stay ahead of emerging threats and vulnerabilities.

At Databricks we continuously monitor the latest trends and research in AI red teaming and integrate them into our lab for use by our operators.

One recent example of this has been the shift towards multi-turn–based techniques such as Microsoft Crescendo, which has proven highly effective across a wide range of models and AI systems.

Testing compound AI systems

Compound AI systems integrate multiple AI subsystems across different modalities, such as language, vision and decision-making. These systems are inherently complex, requiring seamless interaction among distinct components to produce coherent and reliable outputs. Testing them involves assessing the individual modules and interactions between them, ensuring that vulnerabilities in one subsystem don't propagate or create failure points elsewhere.

In our red teaming approach, we emphasize the integrity of individual components and the systemic risks arising from their integration. Cross-modality adversarial testing exploits the dependencies between subsystems, identifying weaknesses that may surface from ambiguous data or conflicting inputs across modalities. For example, a model combining vision and language might perform adequately in isolation but struggle to interpret an image correctly when paired with a related text prompt.

Moreover, interconnected systems and pipelines can be susceptible to traditional software vulnerabilities, necessitating a conventional red team or penetration testing approach. Comprehensive testing of compound AI systems is crucial, as the complexity of multiple interacting components can obscure critical vulnerabilities. Research highlights the importance of targeted adversarial attacks across modalities to ensure robustness across the entire system.

SUPPLY CHAIN

Supply chain attack testing in AI is akin to traditional red teaming and penetration testing, as it focuses on the broader ecosystem surrounding the AI model rather than the model itself. This testing type evaluates the vulnerabilities within the AI development and deployment pipeline — ranging from data acquisition and model training environments to third-party dependencies like libraries, APIs and cloud infrastructure. Just as in traditional cybersecurity red teaming, supply chain attack testing involves simulating real-world attack vectors such as tampering with datasets, injecting malicious code into libraries or compromising the integrity of development tools. The goal is to identify and exploit weaknesses in the supply chain to prevent attackers from manipulating the AI system indirectly, ensuring the overall security and integrity of the model beyond just its functional behavior. This type of testing is crucial for defending against threats that arise from the broader context in which the AI operates.

Conclusion

AI red teaming is essential to ensure the alignment and safety of AI models across various contexts. Our approach blends automated tools for comprehensive and scalable assessment with manual exploration to uncover deeper and more nuanced vulnerabilities. This layered strategy helps ensure that AI models remain robust, safe and aligned with ethical standards, regardless of their modality or the scope of the engagement.

Databricks approach to incident response

AI and machine learning (ML) advancements present significant risks, including data breaches, model manipulation, biased outcomes and regulatory noncompliance. At Databricks, we recognize that any central system reliant on collecting vast amounts of valuable data will inevitably become a prime target for malicious actors. Consequently, an incident response plan (IRP), which includes processes for AI/ML cybersecurity and privacy events, is essential for effectively addressing these challenges.

Our Security Incident Response Team (SIRT) navigates this rapidly evolving landscape with a core mission to protect our customers, employees and enterprise data in a fast, efficient and standardized manner.

The IRP at Databricks consists of several key phases to ensure a structured and effective response to AI and ML security and privacy events. Examples of where Databricks has integrated changes for AI/ML include:

1. **Incident readiness**
   - Investigation acceptance matrix: Including predefined AI-focused requirements in the existing set of requirements designed to focus the goal of the SIRT team on the most risky and impactful issues, typically addressing security, privacy and/or legal concerns
2. **Detection**
   - Monitoring tools: Ingests alerts from monitoring tools to detect anomalies in AI/ML operations such as unexpected model outputs or unauthorized data access

Defining the AI investigation acceptance criteria

Engagement in the incident response process can be a significant investment for the company. It should be primarily reserved for predetermined criteria that substantially impact the overall security health. Many organizations integrating AI or ML products will do so within their existing services, such as embedding a GenAI assistant into a preexisting service. A comprehensive AI-incident response plan (AI-IRP) addresses gaps in the existing IRP. At Databricks, SIRT maintains a continuous 24/7/365 on-call schedule for rapid triage, investigation and mitigation to generate stable outcomes in the following areas:

1. Product security investigation and response (PSIRTs)
2. Corporate security investigation and response (CSIRTs)
3. Data and privacy investigations

With the introduction of the AI threats, the Databricks Security team has updated the IRP with an AI incident response matrix aimed to address three key areas:

1. Security risk: Incidents involving the impact of the components or code
2. Privacy risk: Incidents impacting training data, model data, customer fine-tuned data, personal or private data
3. Other: Incidents impacting areas such as confidentiality and intellectual property, or AI/ML incidents that implicate regulated content such as child sexual abuse material (CSAM)

Conclusion

The Databricks approach to designing the IRP for AI/ML cybersecurity and privacy events is centered on our core mission of protecting customer, employee and enterprise data. By addressing gaps in the previous IRP, establishing clear new acceptance criteria and ensuring cross-organizational alignment, the Security Incident Response Team (SIRT) can effectively contribute to the security dialogue across Databricks. This structured plan enables the security team to conduct table tops for the identified risks by our internal product security team, collaborate with the various engineering teams and test incident readiness, thereby enhancing our overall security posture.

Appendix: Glossary

A

ACID transaction: A set of properties — atomicity, consistency, isolation and durability — that ensure reliable processing of database transactions.

Adversarial examples: Modified testing samples that induce misclassification of a machine learning model at deployment time.

Agentic AI: Autonomous AI systems that can make decisions and take actions independently to achieve specific goals. These systems utilize learning capabilities to adapt to their environments and operate with a level of self-direction, reducing the need for constant human intervention.

AI agent: While the industry is refining the definition of AI agents, generally an AI agent is an application capable of making decisions based on data, learning from experience and adapting to new situations over time. Unlike traditional rule-based systems like robotic process automation (RPA), AI agents can manage structured, unstructured and other data types, analyze them and make informed decisions in dynamic or uncertain environments.

AI augmentation: The use of AI to enhance human capabilities and decision-making rather than replacing them. By automating routine tasks and providing data-driven insights, AI augmentation allows individuals to focus on more strategic activities, improving efficiency and overall outcomes.

AI gateway: A centralized system that acts as an intermediary between users or applications and various AI services, facilitating access through standardized APIs. It manages data routing, ensures security and access control, and provides monitoring and analytics to optimize the use of AI capabilities.

AI governance: The actions to ensure stakeholder needs, conditions and options are evaluated to determine balanced, agreed-upon enterprise objectives; setting direction through prioritization and decision-making; and monitoring performance and compliance against agreed-upon directions and objectives. AI governance may include policies on the nature of AI applications developed and deployed versus those limited or withheld.

AI guardrail: A safeguard that is put in place to prevent AI from causing harm. AI guardrails are a lot like highway guardrails — they're both created to keep people safe and guide positive outcomes.

AI red teaming: The practice of rigorously testing AI systems by simulating adversarial attacks and identifying vulnerabilities. This approach helps organizations improve the security, robustness and ethical safeguards of their AI models by assessing how they respond to malicious or unexpected behaviors.

Artificial intelligence (AI): A multidisciplinary field of computer science that aims to create systems capable of emulating and surpassing human-level intelligence.

Attribute-based access control (ABAC): A security model that grants or denies access to resources based on the attributes of users, resources and the environment. This approach allows for fine-grained access control by evaluating various characteristics such as user roles, resource classifications and contextual factors to determine permissions dynamically.

Autonomous agents: AI systems capable of performing tasks or making decisions independently, without human intervention, by perceiving their environment and acting based on predefined goals or learned behaviors.

B

Bug bounty program: A program that offers monetary rewards to ethical hackers for successfully discovering and reporting a vulnerability or bug to the application's developer. Bug bounty programs allow companies to leverage the hacker community to improve their systems' security posture over time.

C

Compound AI system: An advanced AI architecture that integrates multiple AI models or components to achieve more complex and capable functionalities than a single model can provide.

Compute plane: Where your data is processed in Databricks Platform architecture.

Concept drift: A situation where statistical properties of the target variable change and the very concept of what you are trying to predict changes as well. For example, the definition of what is considered a fraudulent transaction could change over time as new ways are developed to conduct such illegal transactions. This type of change will result in concept drift.

Continuous integration and continuous delivery / continuous deployment (CI/CD): CI is a modern software development practice in which incremental code changes are made frequently and reliably. CI/CD is common to software development, but it is becoming increasingly necessary to data engineering and data science. By automating the building, testing and deployment of code, development teams are able to deliver releases more frequently and reliably than with the manual processes still common to data engineering and data science teams.

Control plane: The back-end services that Databricks manages in your Databricks account. Notebook commands and many other workspace configurations are stored in the control plane and encrypted at rest.

D

Databricks Delta Live Tables: A declarative framework for building reliable, maintainable and testable data processing pipelines. You define the transformations to perform on your data and Delta Live Tables manages task orchestration, cluster management, monitoring, data quality and error handling.

Databricks Feature Store: A centralized repository that enables data scientists to find and share features and also ensures that the same code used to compute the feature values is used for model training and inference.

DatabricksIQ: The data intelligence engine powering the Databricks Platform. It's a compound AI system that combines the use of AI models, retrieval, ranking and personalization systems to understand the semantics of your organization's data and usage patterns.

Databricks Secrets: Sometimes accessing data requires that you authenticate to external data sources through Java Database Connectivity (JDBC). Databricks Secrets stores your credentials so you can reference them in notebooks and jobs instead of directly entering your credentials into a notebook.

Databricks SQL: The collection of services that bring data warehousing capabilities and performance to your existing data lakes. Databricks SQL supports open formats and standard ANSI SQL. An in-platform SQL editor and dashboarding tools allow team members to collaborate with other Databricks users directly in the workspace. Databricks SQL also integrates with a variety of tools so that analysts can author queries and dashboards in their favorite environments without adjusting to a new platform.

Databricks Workflows: Orchestrates data processing, machine learning and analytics pipelines on the Databricks Data Intelligence Platform. Workflows has fully managed orchestration services integrated with the Databricks Platform, including Databricks Jobs to run non-interactive code in your Databricks workspace and Delta Live Tables to build reliable and maintainable ETL pipelines.

Data classification: A crucial part of data governance that involves organizing and categorizing data based on its sensitivity, value and criticality.

Data clean room: A secure environment where multiple parties can share and analyze their data without directly exposing the underlying raw data to each other.

Data drift: The features used to train a model are selected from the input data. When statistical properties of this input data change, it will have a downstream impact on the model's quality. For example, data changes due to seasonality, personal preference changes, trends, etc., will lead to incoming data drift.

Data governance: Data governance is a comprehensive approach that comprises the principles, practices and tools to manage an organization's data assets throughout their lifecycle. By aligning data-related requirements with business strategy, data governance provides superior data management, quality, visibility, security and compliance capabilities across the organization. Implementing an effective data governance strategy allows companies to make data easily available for data-driven decision-making while safeguarding their data from unauthorized access and ensuring compliance with regulatory requirements.

Data Intelligence Platform: A new era of data platform that employs AI models to deeply understand the semantics of enterprise data. It builds on the foundation of the data lakehouse — a unified system to query and manage all data across the enterprise — but automatically analyzes both the data (contents and metadata) and how it is used (queries, reports, lineage, etc.) to add new capabilities.

Data lake: A central location that holds a large amount of data in its native, raw format. Compared to a hierarchical data warehouse, which stores data in files or folders, a data lake uses a flat architecture and object storage to store the data. With object storage, data is stored with metadata tags and a unique identifier, which makes it easier to locate and retrieve data across regions and improves performance. By leveraging inexpensive object storage and open formats, data lakes enable many applications to take advantage of the data.

Data lakehouse: A new, open data management architecture that combines the flexibility, cost-efficiency and scale of data lakes with the data management and ACID transactions of data warehouses, enabling business intelligence (BI) and machine learning (ML) on all data.

Data lineage: A powerful tool that helps organizations ensure data quality and trustworthiness by providing a better understanding of data sources and consumption. It captures relevant metadata and events throughout the data's lifecycle, providing an end-to-end view of how data flows across an organization's data estate.

Data partitioning: A partition is composed of a subset of rows in a table that share the same value for a predefined subset of columns called the partitioning columns. Data partitioning can speed up queries against the table as well as data manipulation.

Data pipeline: A data pipeline implements the steps required to move data from source systems, transform that data based on requirements, and store the data in a target system. A data pipeline includes all the processes necessary to turn raw data into prepared data that users can consume. For example, a data pipeline might prepare data so data analysts and data scientists can extract value from the data through analysis and reporting. An extract, transform and load (ETL) workflow is a common example of a data pipeline.

Data poisoning: Attacks in which a part of the training data is under the control of the adversary.

Data preparation (data prep): The set of preprocessing operations performed in the early stages of a data processing pipeline, i.e., data transformations at the structural and syntactical levels.

Data privacy: Attacks against machine learning models to extract sensitive information about training data.

Data streaming: Data that is continuously and/or incrementally flowing from a variety of sources to a destination to be processed and analyzed in near real-time. This unlocks a new world of use cases around real-time ETL, real-time analytics, real-time ML and real-time operational applications that in turn enable faster decision-making.

Datasets: A dataset in machine learning and artificial intelligence refers to a collection of data that is used to train and test algorithms and models.

Delta Lake: The optimized storage layer that provides the foundation for storing data and tables in the Databricks lakehouse. Delta Lake is open source software that extends Parquet data files with a file-based transaction log for ACID transactions and scalable metadata handling. Delta Lake is fully compatible with Apache Spark™ APIs, and was developed for tight integration with Structured Streaming, allowing you to easily use a single copy of data for both batch and streaming operations and providing incremental processing at scale.

Denial of service (DoS): An attack meant to shut down access to information systems, devices or other network resources, making them inaccessible to their intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e., employees, members or account holders) of the service or resource they expected due to the actions of a malicious cyberthreat actor.

DevSecOps: Stands for development, security and operations. It's an approach to culture, automation and platform design that integrates security as a shared responsibility throughout the entire IT lifecycle.

E

Egress control: Security measures and protocols designed to manage and monitor the exit of individuals and data from a secure environment.

Embeddings: Mathematical representations of the semantic content of data, typically text or image data. Embeddings are generated by a large language model and are a key component of many GenAI applications that depend on finding documents or images that are similar to each other. Examples are RAG systems, recommender systems, and image and video recognition.

Explainable AI: AI systems designed to provide clear, understandable justifications for their predictions or decisions, making it easier for humans to interpret how and why the AI reached its outcomes.

Exploratory data analysis (EDA): Methods for exploring datasets to summarize their main characteristics and identify any problems with the data. Using statistical methods and visualizations, you can learn about a dataset to determine its readiness for analysis and inform what techniques to apply for data preparation. EDA can also influence which algorithms you choose to apply for training ML models.

External models: Third-party models hosted outside of Databricks. Supported by Model Serving, external models allow you to streamline the usage and management of various large language model (LLM) providers, such as OpenAI and Anthropic, within an organization.

Extract, transform and load (ETL): The foundational process in data engineering of combining data from multiple sources into a large, central repository called a data warehouse. ETL uses a set of business rules to clean and organize raw data and prepare it for storage, data analytics and machine learning (ML).

F

Feature engineering: The process of extracting features (characteristics, properties, attributes) from raw data to develop machine learning models.

Fine-tuned LLM: Adapting a pretrained LLM to specific datasets or domains.

Foundation Model: A general purpose machine learning model trained on vast quantities of data and fine-tuned for more specific language understanding and generation tasks.

G

Generative: Type of machine learning methods that learn the data distribution and can generate new examples from distribution.

Generative AI: Also known as GenAI, this is a form of machine learning that uses large quantities of data to train models to produce content.

H

Hallucination: A response generated by AI which contains false or misleading information presented as fact. This term draws a loose analogy with human psychology, where hallucination typically involves false perceptions. However, there's a key difference: AI hallucination is associated with erroneous responses or beliefs rather than perceptual experiences.

Hardened runtime: Databricks handles the actual base system image (e.g., AMI) by leveraging Ubuntu with a hardening configuration based on CIS. As a part of the Databricks Threat and Vulnerability Management program, we perform weekly scanning of the AMIs as they are making their way from dev to production.

Human-in-the-loop (HITL): The process of machine learning that allows people to validate a machine learning model's predictions as right or wrong at the time of training and inference with intervention.

Human-at-the-helm: Cyberattacks that exploit the reliance on human oversight in decision-making, often through misinformation, social engineering or psychological pressure to mislead or manipulate the person in control.

Hybrid models: These models combine multiple methodologies or techniques, often integrating both traditional statistical approaches and modern machine learning methods, to leverage the strengths of each for improved performance and accuracy.

Hyperparameter: A parameter whose value is set before the machine learning process begins. In contrast, the values of other parameters are derived via training.

I

Identity provider (IdP): A service that stores and manages digital identities. Companies use these services to allow their employees or users to connect with the resources they need. They provide a way to manage access, adding or removing privileges, while security remains tight.

Incident response plan (IRP): A documented strategy outlining the processes and procedures an organization follows to prepare for, detect, respond to and recover from cybersecurity incidents.

Inference: The stage of ML in which a model is applied to a task by running data points into a machine learning model to calculate an output such as a single numerical score. For example, a classifier model produces the classification of a test sample.

Inference queries: Inputs sent to an AI model to generate predictions or outputs based on learned patterns and knowledge. In machine learning, inference queries are used after the model is trained, enabling it to make decisions or provide insights in real-world applications without further updating its internal parameters.

Inference tables: A table that automatically captures incoming requests and outgoing responses for a model serving endpoint and logs them as a table.

Initial Access: Techniques that adversaries use with various entry vectors to gain their initial foothold within the system

Insider risk: An insider is any person who has or had authorized access to or knowledge of an organization's resources, including personnel, facilities, information, equipment, networks and systems. Should an individual choose to act against the organization, with their privileged access and their extensive knowledge, they are well positioned to cause serious damage.

Interactive agents: AI systems designed to engage in dynamic, two-way communication with users, often using natural language processing and machine learning to understand and respond to queries or commands.

IP access list (IP ACL): Enables you to restrict access to your AI system based on a user's IP address. For example, you can configure IP access lists to allow users to connect only through existing corporate networks with a secure perimeter. If the internal VPN network is authorized, users who are remote or traveling can use the VPN to connect to the corporate network. If a user attempts to connect to the AI system from an insecure network, like from a coffee shop, access is blocked.

J

Jailbreaking: An attack that employs prompt injection to specifically circumvent the safety and moderation features placed on LLMs by their creators.

L

Label-flipping (LF) attacks: A targeted poisoning attack where the attackers poison their training data by flipping the labels of some examples from one class (i.e., the source class) to another (i.e., the target class).

Lakehouse Monitoring: Databricks Lakehouse Monitoring lets you monitor the statistical properties and quality of the data in all of the tables in your account. You can also use it to track the performance of machine learning models and model serving endpoints by monitoring inference tables that contain model inputs and predictions.

Large language model (LLM): A model trained on massive datasets to achieve advanced language processing capabilities based on deep learning neural networks.

Liquid clustering: A dynamic data clustering approach that adapts to changes in data over time, allowing clusters to evolve and reconfigure as new information becomes available. This method enables more accurate and timely insights by continuously refining groupings based on real-time data inputs, making it suitable for applications where data characteristics frequently shift.

LLM-as-a-judge: A scalable and explainable way to approximate human preferences, which are otherwise very expensive to obtain. Evaluating large language model (LLM) based chat assistants is challenging due to their broad capabilities and the inadequacy of existing benchmarks in measuring human preferences. LLMs as judges to evaluate these models on more open-ended questions.

LLM hallucination: A phenomenon wherein a large language model (LLM) — often a generative AI chatbot or computer vision tool — perceives patterns or objects that are nonexistent or imperceptible to human observers, creating outputs that are nonsensical or altogether inaccurate.

M

Machine learning (ML): A form of AI that learns from existing data and makes predictions without being explicitly programmed.

Machine learning algorithms: Pieces of code that help people explore, analyze and find meaning in complex datasets. Each algorithm is a finite set of unambiguous step-by-step instructions that a machine can follow to achieve a certain goal. In a machine learning model, the goal is to establish or discover patterns that people can use to make predictions or categorize information.

Machine learning models: Process of using mathematical models of data to help a computer learn without direct instruction. Machine learning uses algorithms to identify patterns within data, and those patterns are then used to create a data model that can make predictions. For example, in natural language processing, machine learning models can parse and correctly recognize the intent behind previously unheard sentences or combinations of words. In image recognition, a machine learning model can be taught to recognize objects — such as cars or dogs. A machine learning model can perform such tasks by having it "trained" with a large dataset. During training, the machine learning algorithm is optimized to find certain patterns or outputs from the dataset, depending on the task. The output of this process — often a computer program with specific rules and data structures — is called a machine learning model.

Machine learning operations (MLOps): The practice of creating new machine learning (ML) models and running them through a repeatable, automated workflow that deploys them to production. An MLOps pipeline provides a variety of services to data science processes, including model version control, continuous integration and continuous delivery (CI/CD), model catalogs for models in production, infrastructure management, monitoring of live model performance, security, and governance. MLOps is a collaborative function, often comprising data scientists, devops engineers, security teams and IT.

Malicious libraries: Software components that were intentionally designed to cause harm to computer systems or the data they process. Such packages can be distributed through various means, including phishing emails, compromised websites or even legitimate software repositories.

Metadata: Data that annotates other data and AI assets. It generally includes the permissions that govern access to them with descriptive information, possibly including their data descriptions, data about data ownership, access paths, access rights and data volatility.

MLflow Model Registry: A centralized model store, set of APIs, and UI to collaboratively manage the full lifecycle of an MLflow model. It provides model lineage (which MLflow experiment and run produced the model), model versioning, model aliasing, model tagging and annotations.

MLSecOps: The integration of security practices and considerations into the ML development and deployment process. This includes ensuring the security and privacy of data used to train and test models, as well as protecting deployed models and the infrastructure they run on from malicious attacks.

Model cards: Standardized documentation for machine learning models that provides essential details about a model's purpose, performance and limitations.

Model drift: The decay of models' predictive power as a result of the changes in real-world environments.

Model inference: The use of a trained model on new data to create a result.

Model inversion: In machine learning models, private assets like training data, features and hyperparameters, which are typically confidential, can potentially be recovered by attackers through a process known as model inversion. This technique involves reconstructing private elements without direct access, compromising the model's security.

Model management: A single place for development, tracking, discovering, governing, encrypting and accessing models with proper security controls.

Model operations: The building of predictive ML models, the acquisition of models from a model marketplace, or the use of LLMs like OpenAI or Foundation Models APIs. Developing a model requires a series of experiments and a way to track and compare the conditions and results of those experiments.

Mosaic AI AutoML: Helps you automatically apply machine learning to a dataset. You provide the dataset and identify the prediction target, while AutoML prepares the dataset for model training. AutoML then performs and records a set of trials that creates, tunes and evaluates multiple models. After model evaluation, AutoML displays the results and provides a Python notebook with the source code for each trial run so you can review, reproduce and modify the code. AutoML also calculates summary statistics on your dataset and saves this information in a notebook that you can review later.

Mosaic AI Model Serving: A unified service for deploying, governing, querying and monitoring models fine-tuned or pre-deployed by Databricks like Llama 3, MosaicML MPT or BGE, or from any other model provider like Azure OpenAI, AWS Bedrock, AWS SageMaker and Anthropic. Model Serving provides a highly available and low-latency service for deploying models. The service automatically scales up or down to meet demand changes, saving infrastructure costs while optimizing latency performance.

Mosaic AI Vector Search: A vector database that is built into the Databricks Data Intelligence Platform and integrated with its governance and productivity tools. A vector database is a database that is optimized to store and retrieve embeddings. Embeddings are mathematical representations of the semantic content of data, typically text or image data. Embeddings are generated by a large language model and are a key component of many GenAI applications that depend on finding documents or images that are similar to each other. Examples are RAG systems, recommender systems, and image and video recognition.

Model theft: Theft of a system's knowledge through direct observation of its input and output observations, akin to reverse engineering. This can lead to unauthorized access, copying or exfiltration of proprietary models, resulting in economic losses, eroded competitive advantage and exposure of sensitive information.

Model Zoo: A repository or library that contains pretrained models for various machine learning tasks. These models are trained on large datasets and are ready to be deployed or fine-tuned for specific tasks.

N

Notebook: A common tool in data science and machine learning for developing code and presenting results.

O

Observability: The ability to monitor, understand and gain insights into the health and performance of data pipelines and systems. It involves tracking metrics like data quality, lineage, latency and dependencies, helping teams proactively detect and address issues in data workflows to ensure reliability and trustworthiness across the data lifecycle.

Offline system: ML systems that are trained up, "frozen," and then operated using new data on the frozen trained system.

Online system: An ML system is said to be "online" when it continues to learn during operational use, modifying its behavior over time.

Ontology: A formally defined vocabulary for a particular domain of interest used to capture knowledge about that (restricted) domain of interest. Adversaries may discover the ontology of a machine learning model's output space — for example, the types of objects a model can detect. The adversary may discover the ontology by repeated queries to the model, forcing it to enumerate its output space. Or the ontology may be discovered in a configuration file or in documentation about the model.

P

Penetration testing (pen testing): A security exercise where a cybersecurity expert attempts to find and exploit vulnerabilities in a computer system through a combination of an in-house offensive security team, qualified third-party penetration testers and a year-round public bug bounty program. The purpose of this simulated attack is to identify any weak spots in a system's defenses that attackers could take advantage of.

Predictive optimization: A process that uses predictive analytics and machine learning algorithms to analyze historical data and forecast future outcomes, enabling organizations to make data-driven decisions that enhance performance and efficiency.

Pretrained LLM: Training an LLM from scratch using your own data for better domain performance.

Private link: Enables private connectivity between users and their Databricks workspaces and between clusters on the compute plane and core services on the control plane within the Databricks workspace infrastructure.

Prompt injection

- *Direct:* A direct prompt injection occurs when a user injects text that is intended to alter the behavior of the LLM
- *Indirect:* When a user might modify or exfiltrate resources (e.g., documents, web pages) that will be ingested by the GenAI model at runtime via the RAG process

Q

Query federation: A technique that allows users to execute a single query across multiple data sources, like databases or data lakes, without consolidating the data into one place.

R

Red teaming: NIST defines cybersecurity red teaming as "a group of people authorized and organized to emulate a potential adversary's attack or exploitation capabilities against an enterprise's security posture. The Red Team's objective is to improve enterprise cybersecurity by demonstrating the impacts of successful attacks and by demonstrating what works for the defenders (i.e., the Blue Team) in an operational environment." (CNSS 2015 [80]) Traditional red teaming might combine physical and cyberattack elements, attack multiple systems, and aim to evaluate the overall security posture of an organization. Penetration testing (pen testing), in contrast, tests the security of a specific application or system. In AI discourse, red teaming has come to mean something closer to pen testing, where the model may be rapidly or continuously tested by a set of evaluators and under conditions other than normal operation.

Reinforcement learning from human feedback (RLHF): A method of training AI models where human feedback is used as a source of reinforcement signals. Instead of relying solely on predefined reward functions, RLHF incorporates feedback from humans to guide the learning process.

Resource control: A capability in which the attacker has control over the resources consumed by an ML model, particularly for LLMs and RAG applications.

Responsible AI: Responsible Artificial Intelligence (Responsible AI) is an approach to developing, assessing and deploying AI systems in a safe, trustworthy and ethical way. Characteristics of trustworthy AI systems include: valid and reliable, safe, secure and resilient, accountable and transparent, explainable and interpretable, privacy-enhanced, and fair with harmful bias managed.

Retrieval augmented generation (RAG): An architectural approach that can improve the efficacy of large language model (LLM) applications by leveraging custom data. This is done by retrieving data/documents relevant to a question or task and providing them as context for the LLM.

S

Serverless compute: An architectural design that follows infrastructure as a service (IaaS) and platform as a service (PaaS), and which primarily requires the customer to provide the necessary business logic for execution. Meanwhile, the service provider takes care of infrastructure management. Compared to other platform architectures like PaaS, Serverless provides a considerably quicker path to realizing value and typically offers better cost efficiency and performance.

Single-sign on (SSO): A user authentication tool that enables users to securely access multiple applications and services using just one set of credentials.

Software development lifecycle (SDLC): A structured process that enables the production of high-quality, low-cost software, in the shortest possible production time. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands. The SDLC defines and outlines a detailed plan with stages, or phases, that each encompasses their own process and deliverables. Adherence to the SDLC enhances development speed and minimizes project risks and costs associated with alternative methods of production.

Source code control: A capability in which the attacker has control over the source code of the machine learning algorithm.

Synthetic data generation: Creation of artificial data that mimics real-world data characteristics and structures, typically using algorithms or simulation techniques. This approach is often used to augment training datasets for machine learning models, enhance privacy by obfuscating sensitive information and facilitate testing and validation processes without relying on real data.

System for Cross-domain Identity Management (SCIM): An open standard designed to manage user identity information. SCIM provides a defined schema for representing users and groups, and a RESTful API to run CRUD operations on those user and group resources. The goal of SCIM is to securely automate the exchange of user identity data between your company's cloud applications and any service providers, such as enterprise SaaS applications.

T

Table tops: Simulated, discussion-based sessions designed to help organizations prepare for and respond to potential incidents, typically in the context of emergency management or cybersecurity.

Train proxy: The ability of an attacker to extract training data of a generative model by prompting the model on specific inputs.

Train proxy via replication: Adversaries may replicate a private model. By repeatedly querying the victim's ML Model Inference API Access, the adversary can collect the target model's inferences into a dataset. The inferences are used as labels for training a separate model offline that will mimic the behavior and performance of the target model.

Trojan: A malicious code/logic inserted into the code of a software or hardware system, typically without the knowledge and consent of the organization that owns/develops the system, and which is difficult to detect and may appear harmless, but can alter the intended function of the system upon a signal from an attacker to cause a malicious behavior desired by the attacker. For Trojan attacks to be effective, the trigger must be rare in the normal operating environment so that it does not affect the normal effectiveness of the AI and raise the suspicions of human users.

Trojan horse backdoor: In the context of adversarial machine learning, the term “backdoor” describes a malicious module injected into the ML model that introduces some secret and unwanted behavior. This behavior can then be triggered by specific inputs, as defined by the attacker.

U

Unity Catalog (UC): A unified governance solution for data and AI assets on the Databricks Data Intelligence Platform. It provides centralized access control, auditing, lineage and data discovery capabilities across Databricks workspaces.

V

Vector database: A specialized type of database designed to store and manage vector embeddings, which are high-dimensional representations of data such as text, images or audio. These databases enable efficient similarity searches and retrieval based on the geometric proximity of vectors, making them particularly useful for applications in machine learning, natural language processing and recommendation systems.

Vision models: AI systems that analyze and interpret visual data from images or videos, often using techniques from computer vision and deep learning. These models can perform tasks such as object detection, image classification and facial recognition, enabling applications in fields like autonomous vehicles, medical imaging and security surveillance.

Vulnerability management: An information security continuous monitoring (ISCM) process of identifying, evaluating, treating and reporting on security vulnerabilities in systems and the software that runs on them. This, implemented alongside other security tactics, is vital for organizations to prioritize possible threats and minimizing their "attack surface."

W

Watering hole attacks: A form of cyberattack that targets groups of users by infecting websites that they commonly visit to gain access to the victim's computer and network.

Webhooks: Enable you to listen for Model Registry events so your integrations can automatically trigger actions. You can use webhooks to automate and integrate your machine learning pipeline with existing CI/CD tools and workflows. For example, you can trigger CI builds when a new model version is created or notify your team members through Slack each time a model transition to production is requested.

09 License