databricks

Whitepaper

# Databricks AI Security Framework (DASF)

Version 1.1

databricks

# Table of Contents



## Authors


**Omar Khawaja**
Vice President and Field Chief Information Security Officer
databricks

**Arun Pamulapati**
Senior Staff Security Field Engineer
databricks

**Kelly Albano**
Product Marketing Manager
databricks

# 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 artificial intelligence (AI) technologies for opportunities while managing potential security and privacy risks, such as data breaches and regulatory compliance.**

Adopting AI also raises regulatory considerations, exemplified by President Joe Biden's Executive Order (E.O. 14110) and NIST's AI Risk Management Framework, underlining the importance of responsible governance and oversight. The evolving legal and regulatory landscape, combined with uncertainties around ownership accountability, leaves data, IT and security leaders navigating how to effectively harness generative AI for organizational benefits while addressing perceived risks.

The Databricks Security team created the **Databricks AI Security Framework (DASF)** to address the evolving risks associated with the widespread integration of AI globally. Unlike approaches that focus solely on securing models or endpoints, the DASF adopts a comprehensive strategy to mitigate cyber 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 for collaboration between business, IT, data, AI and security teams throughout the AI lifecycle. It addresses the evolving nature of data science from a research-oriented to a project-based discipline, facilitating structured conversations on security threats and mitigations without needing deep expertise crossover. We believe the document will be valuable to security teams, ML practitioners 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 and compliance of specific ML systems.

The DASF walks its readers through the 12 foundational components of a generic data-centric AI system: 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 55 technical security risks that arise across these components and dedicated a chapter describing the specific component, the associated risks 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.

The framework concludes with Databricks' final recommendations 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 Security workshops, please contact dasf@databricks.com. If you are curious about how Databricks approaches security, please visit our Security and Trust Center.

Security & Trust Center

Your data security is our priority

Learn more →

# 01 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 artificial intelligence (AI) applications. Leveraging these technologies presents opportunities but also potential risks. There is a risk of security and privacy breaches, as the data sent to an external large language model (LLM) could be leaked or summarized. Several organizations have even banned the use of ChatGPT due to sensitive enterprise data being sent by users. Organizations are also concerned about potential hazards such as data loss, 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. In October 2023, President Biden issued an Executive Order on safe, secure and trustworthy artificial intelligence, emphasizing the responsible development and use of AI technologies. The National Institute of Standards and Technology (NIST) recently published its 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.

The Databricks Security team 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.

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

The DASF takes a holistic approach to mitigating 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 engineering teams did not 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 have produced will be helpful to three major audience groups:

**Security teams (CISOs, security leaders, DevSecOPs, 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.

**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.

**Governance leaders, risk officers 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.

If you are 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 will 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 is designed for collaborative use throughout the AI lifecycle by data and AI teams and their security counterparts referenced above. The DASF is meant to foster closer collaboration between these teams and improve the overall security of AI systems. The concepts in this document are applicable for all teams, even if they do not use Databricks to build their use cases. That said, we will 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 do not use Databricks will be able to follow along without issue.

First, we suggest that organizations find out what type of AI models are being built or being used. As a guideline, we define model types broadly as the following:

**Predictive ML models.** These are traditional structured data machine learning models trained on your enterprise tabular data. They are typically Python models packaged in the MLflow format. Examples include scikit-learn, XGBoost, PyTorch and Hugging Face transformer models.

**State-of-the-art open models** made available by Foundation Model APIs. These models are curated foundation model architectures that support optimized inference. Base models, like Meta Llama 3.1-405B-Instruct, BGE-Large and Mixtral-8x7B, 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 these models' usage patterns 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-4, Anthropic's Claude and others.

Second, we recommend that organizations identify where in their organization AI systems are being built, the process, and who is 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.

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 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.

To further refine risk identification, we categorize risks by model type: predictive ML models, RAG-LLMs, fine-tuned LLMs, pretrained LLMs, foundation models and external models. Once the relevant risks are identified, teams can determine which controls are applicable from the comprehensive list in Section 3: Understanding Databricks Data Intelligence Platform 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 on the Databricks Data Intelligence Platform, with reference links to relevant documentation provided.

Our experience shows that implementing these guidelines helps customers build secure and functional AI systems. 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 to get started with it.

**When I think about what makes a good accelerator, it’s all about making things smoother, more efficient and fostering innovation. The DASF is a proven and effective tool for security teams to help their partners get the most out of AI. Additionally, it lines up with established risk frameworks like NIST, so it’s not just speeding things up – it’s setting a solid foundation in security work.**

NAVY FEDERAL Credit Union

**Riyaz Poonawala**
Vice President of Information Security

# 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

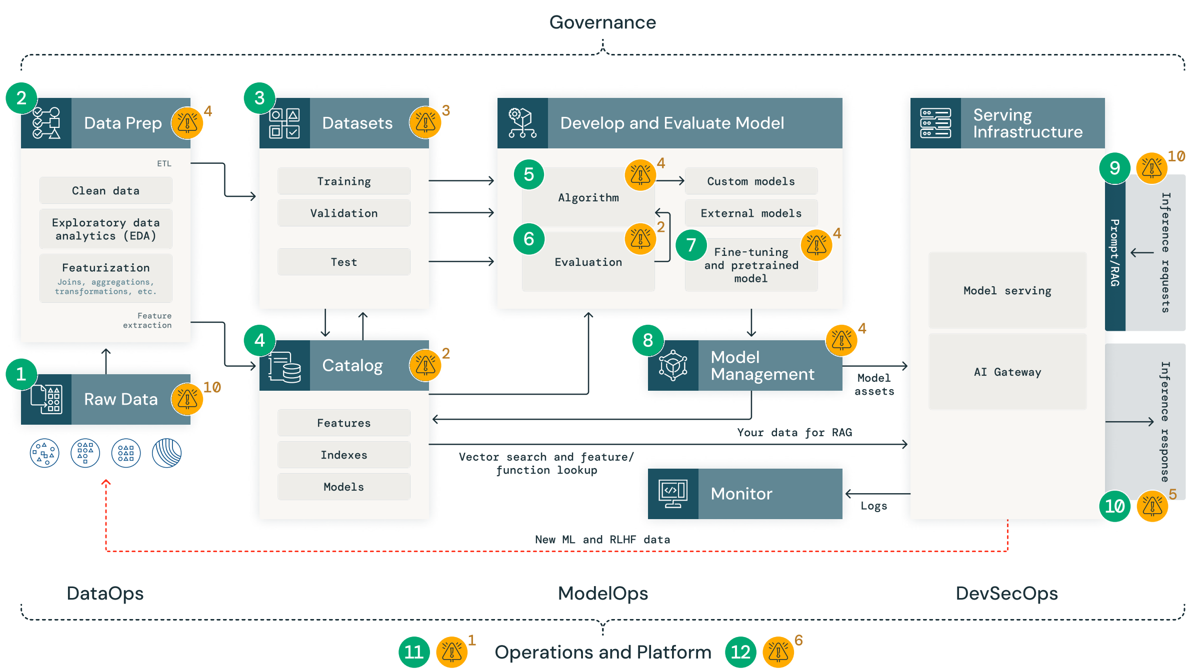



**Figure 1:** Foundational components of a generic data-centric AI system. Numbers in orange indicate risks identified in that specific system.

**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.

**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.

**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.

**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.

In our analysis of AI systems, we identified 55 technical security risks across the 12 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 CISO 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.

| SYSTEM STAGE | SYSTEM COMPONENTS (FIGURE 1) | POTENTIAL SECURITY RISKS |
| --- | --- | --- |
| Data operations | 1 Raw data →<br>2 Data preparation →<br>3 Datasets →<br>4 Catalog and governance → | 19 specific risks:<br>1.1 Insufficient access controls →<br>1.2 Missing data classification →<br>1.3 Poor data quality →<br>1.4 Ineffective storage and encryption →<br>1.5 Lack of data versioning →<br>1.6 Insufficient data lineage →<br>1.7 Lack of data trustworthiness →<br>1.8 Data legal →<br>1.9 Stale data →<br>1.10 Lack of data access logs →<br><br>2.1 Preprocessing integrity →<br>2.2 Feature manipulation →<br>2.3 Raw data criteria →<br>2.4 Adversarial partitions →<br><br>3.1 Data poisoning →<br>3.2 Ineffective storage and encryption →<br>3.3 Label flipping →<br><br>4.1 Lack of traceability and transparency of model assets →<br>4.2 Lack of end-to-end ML lifecycle → |
| Model operations | 5 ML algorithm →<br>6 Evaluation →<br>7 Model build →<br>8 Model management → | 14 specific risks:<br>5.1 Lack of tracking and reproducibility of experiments →<br>5.2 Model drift →<br>5.3 Hyperparameters stealing →<br>5.4 Malicious libraries →<br><br>6.1 Evaluation data poisoning →<br>6.2 Insufficient evaluation data →<br><br>7.1 Backdoor machine learning/Trojaned model →<br>7.2 Model assets leak →<br>7.3 ML supply chain vulnerabilities →<br>7.4 Source code control attack →<br><br>8.1 Model attribution →<br>8.2 Model theft →<br>8.3 Model lifecycle without HITL →<br>8.4 Model inversion → |

| SYSTEM STAGE | SYSTEM COMPONENTS (FIGURE 1) | POTENTIAL SECURITY RISKS |
|---|---|---|
| Model deployment and serving | 9 Model Serving — inference requests →<br>10 Model Serving — inference responses → | 15 specific risks:<br>9.1 Prompt inject →<br>9.2 Model inversion →<br>9.3 Model breakout →<br>9.4 Looped input →<br>9.5 Infer training data membership →<br>9.6 Discover ML model ontology →<br>9.7 Denial of service (DOS) →<br>9.8 LLM hallucinations →<br>9.9 Input resource control →<br>9.10 Accidental exposure of unauthorized data to models →<br>10.1 Lack of audit and monitoring inference quality →<br>10.2 Output manipulation →<br>10.3 Discover ML model ontology →<br>10.4 Discover ML model family →<br>10.5 Black-box attacks → |
| Operations and platform | 11 ML operations →<br>12 ML platform → | 7 specific risks:<br>11.1 Lack of MLOps — repeatable enforced standards →<br>12.1 Lack of vulnerability management →<br>12.2 Lack of penetration testing and bug bounty →<br>12.3 Lack of incident response →<br>12.4 Unauthorized privileged access →<br>12.5 Poor SDLC →<br>12.6 Lack of compliance → |

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

**Note:** We are 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.

## 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, 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<br>**Insufficient access controls**<br>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.<br>Data operations → | DASF 1 SSO with IdP and MFA to authenticate and limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to authorize access to data<br>DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests<br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>DASF 51 Share data and AI assets securely<br>DASF 59 Use clean rooms to collaborate in a secure environment<br>DASF 55 Monitor audit logs<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |
| RAW DATA 1.2<br>**Missing data classification**<br>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.<br>Data operations → | DASF 6 Classify data with tags as it is ingested into the platform aligning with the organization's governance requirements<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| RAW DATA 1.3<br>**Poor data quality**<br>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.<br>Data operations → | DASF 7 Enforce data quality checks on batch and streaming datasets<br>DASF 21 Monitor data and AI system from a single pane of glass<br>DASF 36 Set up monitoring alerts<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |
| RAW DATA 1.4<br>**Ineffective storage and encryption**<br>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.<br>Data operations → | DASF 8 Encrypt data at rest<br>DASF 9 Encrypt data in transit<br>DASF 5 Control access to data and other objects for metadata encryption across all data assets<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |
| RAW DATA 1.5<br>**Lack of data versioning**<br>When data gets corrupted by a malicious user by introducing a new set of data or by corrupting a data pipeline, you will need to be able to roll back or trace back to the original data.<br>Data operations → | DASF 10 Version data and track change logs on large-scale datasets that are fed to your models<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |
| RAW DATA 1.6<br>**Insufficient data lineage**<br>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.<br>Data operations → | DASF 11 Capture and view data lineage<br>DASF 51 Share data and AI assets securely<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |

| 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: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

RAW DATA 1.8

## Data legal

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 29 Build MLOps workflows to track models and trace data sources and lineage to retrain models with the updated dataset by following legal constraints
- DASF 27 Pretrain a large language model (LLM) to only use the data that is allowed with LLMs for inference

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ● | Fine-tuned LLMs: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

RAW DATA 1.9

## Stale data

When downstream data is not 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 are delivered promptly to the model.

Data operations →

- DASF 13 Use near real-time data for fault-tolerant, near real-time data ingestion, processing and machine learning, and AI for streaming data
- DASF 7 Enforce data quality checks on batch and streaming datasets

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ● | Fine-tuned LLMs: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

RAW DATA 1.10

## Lack of data access logs

Without proper audit mechanisms, an organization may not be fully aware of its risk surface area, leaving it 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

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ● | Fine-tuned LLMs: | ● |
| Pre-trained 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. **Label 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.

> **Companies need not sacrifice security for AI innovation. The Databricks AI Security Framework is a comprehensive tool to enable AI adoption securely. It not only maps AI security concerns to the AI development pipeline, but makes them actionable for Databricks customers with practical controls. We're pleased to have contributed to the development of this valuable community resource.**

ROBUST INTELLIGENCE
Hyrum Anderson
CTO

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| DATA PREP 2.1<br>**Preprocessing integrity**<br>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.<br>Insiders or external actors can introduce errors or manipulate data during preprocessing or from the information repository itself.<br>Data operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests<br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>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<br>DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources<br>DASF 15 Explore datasets and identify problems<br>DASF 52 Source Code Control<br>DASF 16 Secure model features to reduce the risk of malicious actors manipulating the features that feed into ML training<br>DASF 42 Data-centric MLOps and LLMOps promote models as code<br>DASF 55 Monitor audit logs<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |
| DATA PREP 2.2<br>**Feature manipulation**<br>In almost all cases, raw data requires preprocessing and transformation before it is 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).<br>Data operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists to limit IP addresses that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as a strong control that limits the source for inbound requests<br>DASF 16 Secure model features to prevent and track unauthorized updates to features and for lineage or traceability<br>DASF 42 Data-centric MLOps and LLMOps promote models as code<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| DATA PREP 2.3<br>**Raw data criteria**<br>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.<br>Data operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists to restrict the IP addresses that can authenticate to Databricks<br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br>DASF 43 Use access control lists to control access to data, data streams and notebooks<br>DASF 42 Data-centric MLOps and LLMOps for unit and integration testing<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |
| DATA PREP 2.4<br>**Adversarial partitions**<br>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 is 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.<br>Data operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists to restrict the IP addresses that can authenticate to Databricks<br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br>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<br>DASF 42 Data-centric MLOps and LLMOps for unit and integration testing<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |

"

**The DASF is a very important, foundational document. I think it will go far in helping to bridge the knowledge gap between ML and security experts.**

**Protect AI** — **Diana Kelley**, CISO

## 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.

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| DATASETS 3.1<br>**Data poisoning**<br>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.<br>Data operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists to restrict the IP addresses that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>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<br>DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources<br>DASF 16 Secure model features<br>DASF 17 Track and reproduce the training data used for ML model training and identify ML models and runs derived from a particular dataset<br>DASF 51 Share data and AI assets securely<br>DASF 14 Audit actions performed on datasets<br>DASF 55 Monitor audit logs<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| DATASETS 3.2<br>**Ineffective storage and encryption**<br>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.<br>Data operations → | DASF 8 Encrypt data at rest<br>DASF 9 Encrypt data in transit<br>DASF 5 Control access to data and other objects for metadata encryption across all data assets<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |
| DATASETS 3.3<br>**Label flipping**<br>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.<br>Data operations → | DASF 8 Encrypt data at rest<br>DASF 9 Encrypt data in transit<br>DASF 5 Control access to data and other objects for metadata encryption across all data assets<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ○  Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○  Foundational models: ○  External models: ○ |

> **The DASF is a great example of Databricks' leadership in AI and is a valuable contribution to the industry at a critical time. We know the greatest risk associated with artificial intelligence for the foreseeable future is bad people, and this framework offers an effective counterbalance to those cybercriminals. The DASF is a pragmatic, operational and efficient way to secure your organization.**

HIDDENLAYER

**Chris "Tito" Sestito**
CEO and Co-founder

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

## 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.

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| **GOVERNANCE 4.1**<br>**Lack of traceability and transparency of model assets**<br>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:<br>▪ Undermine the supportability and adoption of these systems, as it hampers the ability to maintain and update them effectively<br>▪ Impact trust and transparency, which are essential for users to understand and rely on the system's decisions<br>▪ 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<br>Data operations → | DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>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<br>DASF 11 Capture and view data lineage to capture the lineage all the way to the original raw data sources<br>DASF 16 Secure model features<br>DASF 17 Track and reproduce the training data used for ML model training and identify ML models and runs derived from a particular dataset<br>DASF 18 Govern model assets for traceability<br>DASF 55 Monitor audit logs<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |
| **GOVERNANCE 4.2**<br>**Lack of end-to-end ML lifecycle**<br>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.<br>Data operations → | DASF 19 Manage end-to-end machine learning lifecycle for measuring, versioning, tracking model artifacts, metrics and results<br>DASF 42 Data-centric MLOps and LLMOps unit and integration testing<br>DASF 21 Monitor data and AI system from a single pane of glass<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ● External models: ● |

## 2.5 Machine Learning Algorithms

A machine learning algorithm is a method that operates on a dataset to produce an ML model that optimizes a model task on the data. 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 is crucial to recognize and mitigate certain security risks associated with the choice of algorithm and its operational mode.

Machine learning algorithms primarily fall 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<br><br>**Lack of tracking and reproducibility of experiments**<br><br>ML development is often poorly documented and tracked, and results that cannot be reproduced may lead to overconfidence in an ML system's performance. Common issues include:<br>▪ Critical details missing from a model's description<br>▪ Results that are fragile, producing dramatically different results on a different GPU (even one that is supposed to be spec-identical)<br>▪ 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)<br><br>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.<br><br>Model operations → | DASF 20 Track ML training runs for documenting, measuring, versioning, tracking model artifacts including algorithms, training environment, hyperparameters, metrics and results<br><br>DASF 42 Data-centric MLOps and LLMOps promote models as code and automate ML tasks for cross-environment reproducibility<br><br>DASF 55 Monitor audit logs<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| ALGORITHMS 5.2<br>**Model drift**<br>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:<br>▪ **Concept drift:** where the statistical properties of the target variable change over time<br>▪ **Data drift:** involving changes in the distribution of input data<br>▪ **Upstream data changes:** occur due to alterations in data collection or processing methods before the data reaches the model<br>Clever attackers can exploit these scenarios to evade an ML system for adversarial purposes.<br>Model operations → | DASF 17 Track training data with MLflow and Delta Lake to track upstream data changes<br>DASF 16 Secure model features to track changes to features<br>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.<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |
| ALGORITHMS 5.3<br>**Hyperparameters stealing**<br>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.<br>Model operations → | DASF 20 Track ML training runs in the model development process, including parameter settings, securely<br>DASF 43 Use access control lists via workspace access controls<br>DASF 42 Data-centric MLOps and LLMOps employing separate model lifecycle stages by UC schema<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |
| ALGORITHMS 5.4<br>**Malicious libraries**<br>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).<br>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<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |

## 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 is 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 45 Evaluate models to capture performance insights for language models
- DASF 44 Trigger actions in response to a specific event via automated jobs to notify human-in-the-loop (HITL)
- DASF 49 Automate LLM evaluation
- DASF 42 Data-centric MLOps and LLMOps unit and integration testing

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ● |
| Pre-trained 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 is 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 47 Compare LLM outputs on set prompts to assess LLM project with an interactive prompt interface
- DASF 45 Evaluate models to capture performance insights for language models

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ● |
| Pre-trained 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 backdoors 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 is crucial to carefully consider and mitigate these potential risks before deploying any pretrained model to production.

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL 7.1<br>**Backdoor machine learning/ Trojaned model**<br>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.<br>Model operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 43 Use access control lists to limit who can bring models and limit the use of public models<br>DASF 42 Data-centric MLOps and LLMOps 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 backdoors. See resources section for third-party tools.<br>DASF 23 Register, version, approve, promote and deploy models and scan models for malicious code when using third-party models or libraries<br>DASF 19 Manage end-to-end machine learning lifecycle<br>DASF 5 Control access to data and other objects<br>DASF 34 Run models in multiple layers of isolation. Models are considered untrusted code: deploy models and custom LLMs with multiple layers of isolation.<br>DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses<br>DASF 55 Monitor audit logs<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ○  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ○  Foundational models: ○  External models: ● |

databricks

Executive Summary

Introduction

→ Risks in AI System Components

Understanding the Databricks Data Intelligence Platform

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

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 24 Control access to models and model assets
- 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 42 Data-centric MLOps and LLMOps to maintain separate model lifecycle stages
- DASF 33 Manage credentials securely to prevent credentials of data sources used for model training from leaking through models
- DASF 55 Monitor audit logs

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ● |
| Pre-trained 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 48 Use hardened runtime for machine learning
- DASF 53 Third-party library control
- DASF 42 Data-centric MLOps and LLMOps 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 backdoors. See resources section for third-party tools.
- DASF 45 Evaluate models and validate (aka, stress testing) to verify reported function and disclosed weaknesses in the models
- 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: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

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: | ● |
| Pre-trained 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 are even aware that they are doing so.

Organizations can increase trust by creating a centralized place 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<br>**Model attribution**<br>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.<br>Model operations → | DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>DASF 28 Create model aliases, tags and annotations for documenting and discovering models<br>DASF 29 Build MLOps workflows with human-in-the-loop (HITL) , model stage management and approvals<br>DASF 51 Share data and AI assets securely<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ○  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ●  External models: ● |

“

**Companies need not sacrifice security for AI innovation. The Databricks AI Security Framework is a comprehensive tool supporting the adoption of secure AI. We are grateful for Databricks' partnership in the journey to trustworthy AI and this tool makes AI security practical and actionable for Databricks customers.**

HITRUST

**Robert Booker**
Chief Strategy Officer

databricks

Executive Summary

Introduction

→ Risks in AI System Components

Understanding the Databricks Data Intelligence Platform

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

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 51 Share data and AI assets securely
- 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 59 Use clean rooms to collaborate in a secure environment
- DASF 55 Monitor audit logs

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ● |
| Pre-trained 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

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ● | Fine-tuned LLMs: | ○ |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL MANAGEMENT 8.4<br>**Model inversion**<br>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.<br>Model operations → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>DASF 24 Control access to models and model assets<br>DASF 30 Encrypt models<br>DASF 31 Secure model serving endpoints<br>DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit APIs<br>DASF 55 Monitor audit logs<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ○  Fine-tuned LLMs: ●<br>Pre-trained 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.

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|

MODEL SERVING — INFERENCE REQUESTS 9.1

## Prompt inject

A direct prompt injection occurs when a user injects text that is 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 are typically required to handle untrusted input and it is 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 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs

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 54 Implement LLM guardrails

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

**Additional controls to consider:**
Robust Intelligence AI Firewall Prompt Injection rule: Flags malicious user input that might direct the LLM to perform an action unintended by the model creator.

HiddenLayer AISec SafeLLM Proxy.

Please see the resources section for a collection of third-party tools.

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL SERVING — INFERENCE REQUESTS 9.2<br><br>**Model inversion**<br><br>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 cannot otherwise access. The attacker can also recover a functionally equivalent model by iteratively querying the model.<br><br>Model deployment and serving → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br><br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br><br>DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform<br><br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br><br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br><br>DASF 24 Control access to models and model assets<br><br>DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs<br><br>DASF 30 Encrypt models<br><br>DASF 31 Secure model serving endpoints<br><br>DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.<br><br>Designing robust prompts can help mitigate attacks such as jailbreaking.<br><br>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.<br><br>Open source and commercial solutions provide a variety of modules including prompt and output scanners for various responsible AI or jailbreaking attacks.<br><br>DASF 37 Set up inference tables for monitoring and debugging model prompts<br><br>DASF 54 Implement LLM guardrails<br><br>**Additional controls to consider:**<br>Robust Intelligence AI Firewall Prompt Injection rule: Flags malicious user input that might direct the LLM to perform an action unintended by the model creator.<br><br>Please see the resources section for a collection of third-party tools.<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ○  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
| --- | --- |
| MODEL SERVING — INFERENCE REQUESTS 9.3<br>**Model breakout**<br>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.<br>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.<br>DASF 37 Set up inference tables for monitoring and debugging model prompts<br>DASF 56 Restrict outbound connections from models to prevent attacks to exfiltrate data, inference requests and responses<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |
| MODEL SERVING — INFERENCE REQUESTS 9.4<br>**Looped input**<br>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.<br>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.<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |

“

**As organizations strive to incorporate machine learning and generative AI capabilities, a meticulous approach to security and governance throughout the AI lifecycle is essential. The Databricks AI Security Framework stands as a guiding light, providing actionable control recommendations and fostering collaboration among diverse AI teams. In the dynamic landscape of AI, this framework serves as a comprehensive guide, addressing security risks at every stage of the AI/ML lifecycle, ensuring responsible, secure and compliant integration for the organization.**

**Carnegie Mellon University**

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

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|

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 →

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

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 45 Evaluate models for custom evaluation metrics

DASF 54 Implement LLM guardrails

**Additional controls to consider:**
Robust Intelligence AI Firewall Prompt Injection rule: Flags malicious user input that might direct the LLM to perform an action unintended by the model creator.

Robust Intelligence AI Firewall PII Detection rule: Flags user input and model output suspected of containing PII.

The HiddenLayer AISec Platform, specifically MLDR, monitors inputs and related outputs to ML models to determine if an adversary is attempting an inference with a malicious intent.

Please see the resources section for a collection of third-party tools.

**Applicable AI deployment model:**

| Model | Applicable | Model | Applicable | Model | Applicable |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL SERVING — INFERENCE REQUESTS 9.6<br><br>**Discover ML model ontology**<br><br>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.<br><br>Model deployment and serving → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br><br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br><br>DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform<br><br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br><br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br><br>DASF 24 Control access to models and model assets<br><br>DASF 28 Create model aliases, tags and annotations<br><br>DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs<br><br>DASF 30 Encrypt models<br><br>DASF 31 Secure model serving endpoints<br><br>DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.<br><br>Designing robust prompts can help mitigate attacks such as jailbreaking.<br><br>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.<br><br>Open source and commercial solutions provide a variety of modules including prompt and output scanners for various responsible AI or jailbreaking attacks.<br><br>DASF 37 Set up inference tables for monitoring and debugging model prompts<br><br>DASF 45 Evaluate models for custom evaluation metrics<br><br>DASF 54 Implement LLM guardrails<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL SERVING — INFERENCE REQUESTS 9.7<br><br>**Denial of service (DoS)**<br><br>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 are often expensive bottlenecks that can become overloaded. Adversaries can intentionally craft inputs that require heavy amounts of useless compute from the machine learning system.<br><br>Model deployment and serving → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br><br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br><br>DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform<br><br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br><br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br><br>DASF 24 Control access to models and model assets<br><br>DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs<br><br>DASF 30 Encrypt models<br><br>DASF 31 Secure model serving endpoints<br><br>DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.<br><br>Designing robust prompts can help mitigate attacks such as jailbreaking.<br><br>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.<br><br>DASF 37 Set up inference tables for monitoring and debugging prompts<br><br>**Additional controls to consider:**<br>Robust Intelligence AI Firewall Prompt Injection rule: Flags malicious user input that might direct the LLM to perform an action unintended by the model creator.<br><br>Please see the resources section for a collection of third-party tools.<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ● External models: ● |

| 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 Create embeddings to securely integrate data objects with sensitive data that goes into LLMs

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 LLM guardrails

**Additional controls to consider:**
Use guardrails to define and enforce assurance for LLM applications. Please see the resources section for a collection of third-party tools.

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ○ | RAG-LLMs: | ● | Fine-tuned LLMs: | ● |
| Pre-trained 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 LLM guardrails

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

**Additional controls to consider:**
Robust Intelligence AI Firewall Prompt Injection rule:
Flags malicious user input that might direct the LLM to perform an action unintended by the model creator.

Robust Intelligence AI Firewall PII Detection rule: Flags user input and model output suspected of containing PII.

Please see the resources section for a collection of third-party tools.

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ● | Fine-tuned LLMs: | ○ |
| Pre-trained LLMs: | ○ | Foundational models: | ○ | External models: | ○ |

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL SERVING — INFERENCE REQUESTS 9.10<br>**Accidental exposure of unauthorized data to models**<br>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 is not 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.<br>Model deployment and serving → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br>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<br>DASF 16 Secure model features to reduce the risk of malicious actors manipulating the features that feed into ML training<br>DASF 46 Store and retrieve embeddings securely to integrate data objects for security-sensitive data that goes into LLMs as RAG inputs<br>DASF 58 Protect data with filters and masking<br>DASF 57 Use attribute-based access controls (ABAC)<br>DASF 59 Use clean rooms to collaborate in a secure environment<br>DASF 55 Monitor audit logs<br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ○  External models: ○ |

## 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<br><br>**Lack of audit and monitoring inference quality**<br><br>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.<br><br>These tables should include the model's user or system making the request, inputs, and the corresponding predictions or outputs. Monitoring the model serving endpoints provides real-time audit in operational settings.<br><br>Model deployment and serving → | DASF 35 Track model performance to evaluate quality<br><br>DASF 36 Set up monitoring alerts<br><br>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.<br><br>DASF 55 Monitor audit logs<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ●  External models: ● |
| MODEL SERVING — INFERENCE RESPONSE 10.2<br><br>**Output manipulation**<br><br>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.<br><br>Model deployment and serving → | DASF 30 Encrypt models for model endpoints with encryption in transit<br><br>DASF 31 Secure model serving endpoints<br><br>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.<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ○  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ●  External models: ● |

> **The DASF is the first-ever framework that would allow businesses to mitigate AI/ML risks at scale versus approaches that operate in silos — collectivism at best for responsible AI/ML.**

**Capital One Financial**

**Ebrima N. Ceesay, PhD, CISSP**
Senior Distinguished Engineer

| RISK/DESCRIPTION | MITIGATION CONTROLS |
|---|---|
| MODEL SERVING — INFERENCE RESPONSE 10.3<br><br>**Discover ML model ontology**<br><br>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.<br><br>Model deployment and serving → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br><br>DASF 2 Sync users and groups to inherit your organizational roles to access data to restrict IP addresses<br><br>DASF 3 IP access lists to restrict the IP addresses that can authenticate to Databricks<br><br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br><br>DASF 5 Unity Catalog privileges and securable objects for permissions model across all data assets to protect data and sources<br><br>DASF 24 Protect model assets, lifecycle and security with UC in MLflow Model Registry<br><br>DASF 28 Create and model aliases, tags and annotations in Unity Catalog for documenting and discovering models<br><br>DASF 30 Encrypt models<br><br>DASF 31 Secure serving endpoint with Model Serving<br><br>DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.<br><br>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.<br><br>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.<br><br>DASF 37 Set up inference tables for monitoring and debugging models |

**Applicable AI deployment model:**

| | | | | | |
|---|---|---|---|---|---|
| Predictive ML models: | ● | RAG-LLMs: | ○ | Fine-tuned LLMs: | ○ |
| Pre-trained LLMs: | ● | Foundational models: | ○ | External models: | ○ |

"

**The Databricks AI Security Framework provides a comprehensive set of actionable guidelines to help secure our data and AI ecosystem end to end.**

Barracuda

**Grizel Lopez**
Senior Director of Engineering

| RISK/DESCRIPTION | MITIGATION CONTROLS |
| --- | --- |
| MODEL SERVING — INFERENCE RESPONSE 10.4<br>**Discover ML model family**<br>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.<br>Model deployment and serving → | DASF 1 SSO with IdP and MFA to limit who can access your data and AI platform<br>DASF 2 Sync users and groups to inherit your organizational roles to access data<br>DASF 3 Restrict access using IP access lists that can authenticate to your data and AI platform<br>DASF 4 Restrict access using private link as strong controls that limit the source for inbound requests<br>DASF 5 Control access to data and other objects for permissions model across all data assets to protect data and sources<br>DASF 24 Control access to models and model assets<br>DASF 28 Create model aliases, tags and annotations<br>DASF 46 Store and retrieve embeddings securely to integrate data objects to integrate data data that goes into LLMs as RAG inputs<br>DASF 30 Encrypt models<br>DASF 31 Secure model serving endpoints<br>DASF 32 Streamline the usage and management of various large language model (LLM) providers and rate-limit inference queries allowed by the model.<br>Designing robust prompts can help mitigate attacks such as jailbreaking.<br>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.<br>Open source and commercial solutions provide a variety of modules including prompt and output scanners for various responsible AI or jailbreaking attacks.<br>DASF 37 Set up inference tables for monitoring and debugging models<br>DASF 45 Evaluate models for custom evaluation metrics<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ○<br>Pre-trained LLMs: ○ Foundational models: ○ External models: ○ |
| MODEL SERVING — INFERENCE RESPONSE 10.5<br>**Black-box attacks**<br>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 are notable for requiring significantly less access to the target system. Successful black-box attacks quickly erode trust in enterprises serving the model connectors.<br>Model deployment and serving → | DASF 30 Encrypt models for model endpoints with encryption in transit<br>DASF 31 Secure model serving endpoints<br>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.<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ○ Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ○ External models: ○ |

## 2.11 Machine Learning Operations (MLOps)

MLOps is a useful approach for creating quality AI solutions. It is 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<br><br>**Lack of MLOps — repeatable enforced standards**<br><br>Operationalizing an ML solution requires joining data from predictions, monitoring and feature tables with other relevant data.<br><br>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.<br><br>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.<br><br>Operations and platform → | DASF 45 Evaluate models to capture performance insights for language models<br><br>DASF 44 Trigger actions in response to a specific event to trigger automated jobs to keep human-in-the-loop (HITL)<br><br>DASF 42 Data-centric MLOps and LLMOps. MLOps best practices: separate environments by workspace and schema, promote models with code, MLOps Stacks for repeatable ML infra across environments.<br><br>**Applicable AI deployment model:**<br>Predictive ML models: ●  RAG-LLMs: ●  Fine-tuned LLMs: ●<br>Pre-trained LLMs: ●  Foundational models: ●  External models: ● |

## 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<br>**Lack of vulnerability management**<br>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.<br>Operations and platform → | DASF 38 Platform security — vulnerability management to build, deploy and monitor AI/ML models on a platform that takes responsibility seriously and shares remediation timeline commitments<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ● External models: ○ |
| PLATFORM 12.2<br>**Lack of penetration testing and bug bounty**<br>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.<br>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.<br>Operations and platform → | DASF 39 Platform security — penetration testing and bug bounty 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.<br>**Applicable AI deployment model:**<br>Predictive ML models: ● RAG-LLMs: ● Fine-tuned LLMs: ●<br>Pre-trained LLMs: ● Foundational models: ● External models: ○ |

databricks


Executive Summary

Introduction

→ Risks in AI System Components

Understanding the Databricks Data Intelligence Platform

Understanding Databricks Risk AI Mitigation Controls

Conclusion

Resources and Further Reading

Acknowledgments

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: | ● |
| Pre-trained 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: | ● |
| Pre-trained 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: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ● | External models: | ○ |

PLATFORM 12.6

### Lack of compliance

As AI applications become prevalent, they are 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: | ● |
| Pre-trained LLMs: | ● | Foundational models: | ● | External models: | ○ |

# 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 lakehouse architecture 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. The Data Intelligence Engine, at the heart of the platform, understands the semantics of your data and how it flows across all 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.

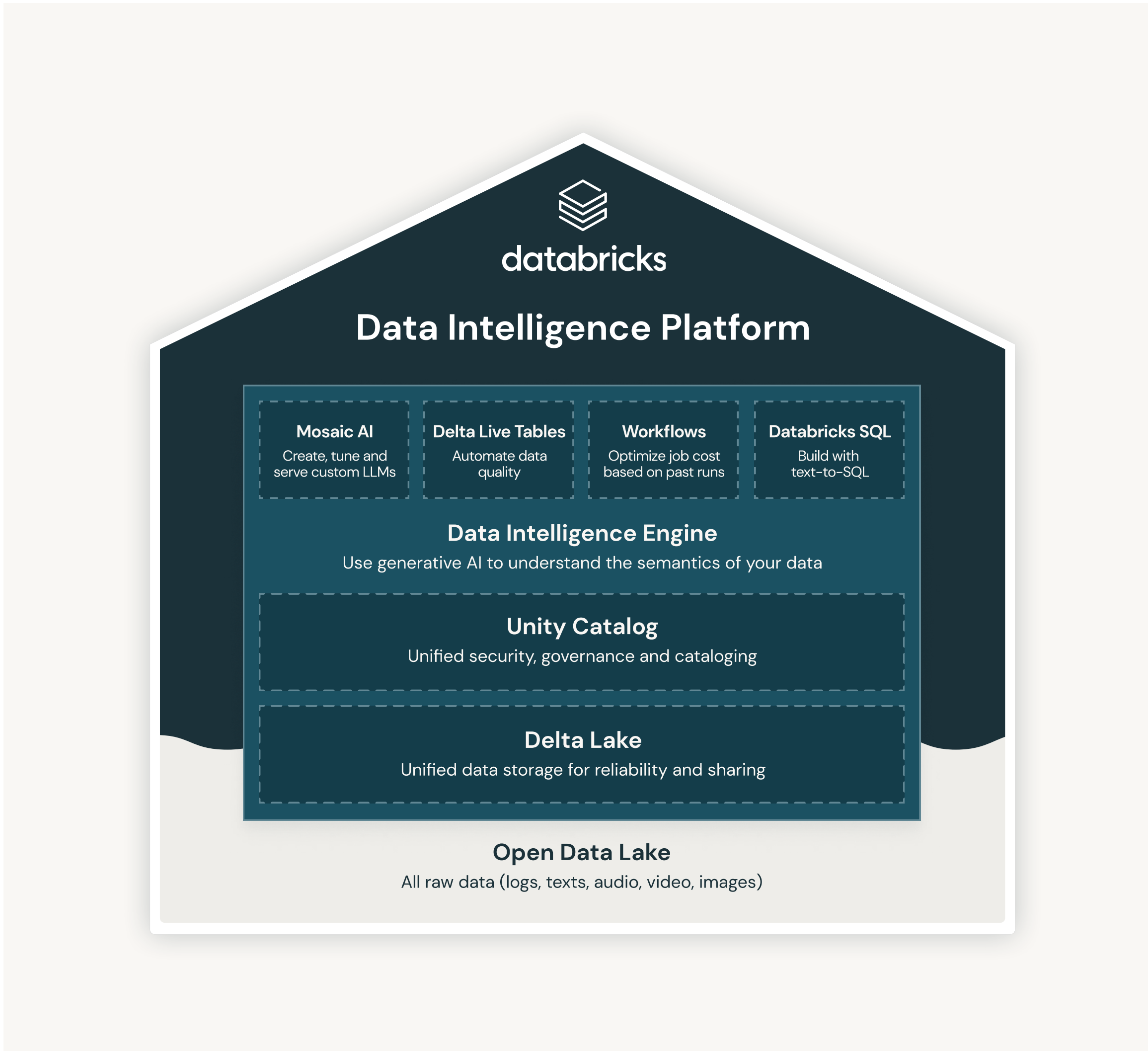
In this section, we provide an overview of our platform and its architecture and components related to governance, security, and AI and machine learning.

MOSAIC AI

DATABRICKS UNITY CATALOG

DATABRICKS PLATFORM ARCHITECTURE

DATABRICKS PLATFORM SECURITY




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.

## 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 both build models from scratch and tune existing models on your data to maintain privacy and control. 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. Our AI offerings include:

1. **End-to-end RAG** (retrieval augmented generation) to build high-quality conversational agents on your data, leveraging the Mosaic AI Vector Search (Public Preview) for increased relevance and accuracy.
2. **Integrating data-centric applications** with leading AI APIs like OpenAI.
3. **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.
4. **Efficient and secure serverless inference** on your enterprise data and connected to Unity Catalog's governance and quality monitoring functionality.
5. **End-to-end MLOps** based on the popular MLflow open source project, with all data produced automatically actionable, tracked and monitorable in the lakehouse.
6. **Improve visibility** and proactively detect anomalies in your entire data and AI workflow, reducing risks, time to value, and high operational costs with Databricks Lakehouse Monitoring (Public Preview).

## Databricks Unity Catalog

Databricks Unity Catalog is the industry's first unified governance solution for data and AI on the lakehouse. With Unity Catalog, organizations can seamlessly govern their structured and unstructured data, machine learning models, notebooks, dashboards, and files on any cloud or platform. Data scientists, analysts and engineers can use Unity Catalog to securely discover, access and collaborate on trusted data and AI assets, leveraging AI to boost productivity and unlock the full potential of the lakehouse environment. This unified approach to governance accelerates data and AI initiatives while ensuring regulatory compliance in a simplified manner. Unity Catalog provides:

1. **Access control for data and AI:** Unity Catalog is the only governance solution for data and AI. The foundational capabilities of Unity Catalog are in governance and access control of all your data and AI assets.  This simplified governance experience works across workspaces and clouds helps you manage your entire data estates. Discover and classify structured and unstructured data, ML models, notebooks, dashboards and arbitrary files on any cloud. Consolidate, map and query data from various platforms, including MySQL, PostgreSQL, Amazon Redshift, Snowflake, Azure SQL, Azure Synapse and Google's BigQuery in one place. Accelerate your data and AI initiatives with a single point of access for data exploration. Boost productivity by securely searching, understanding and extracting insights from your data and AI using natural language.
2. **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.
3. **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.

4 **Automated lineage for all workloads:** Build better understanding of your data estate with automated lineage, tags and auto-generated data insights. Create a unified, transparent view of your entire data ecosystem with automated and granular lineage for all workloads in SQL, R, Python and Scala, and across all asset types — tables, files, notebooks, workflows and dashboards. Lineage can be retrieved via REST APIs to support integrations with our catalog partners.

5 **Security and compliance:** Ability to define access policies at scale for all data and AI assets such as files, tables, ML models, notebooks and dashboards and to audit the access patterns.

## Databricks Platform Architecture

Databricks is a hybrid 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, unlike a pure SaaS, Databricks doesn't know what data your teams process with the Databricks Platform. The actual code, business logic, model artifacts, SaaS, open source models, choice of LLMs, and datasets are provided by your teams. You won't find recommendations like "truncate user IDs" or "hash feature names" because we don't know what data you're analyzing and what models you are deploying.

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.

## 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.

1. **Trust:** Third-party audit firms regularly audit Databricks systems and processes. Databricks customers can trust independent validation of internal security processes.
2. **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.
3. **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.

04

# Understanding Databricks AI Risk Mitigation Controls

In this section, we delve into the comprehensive risk mitigation controls available in the Databricks Data Intelligence Platform for artificial intelligence (AI) and machine learning (ML). As organizations increasingly harness the power of AI, a nuanced understanding of these robust controls becomes imperative to ensure data integrity, security and regulatory compliance throughout the data lifecycle.

At Databricks, we strive to continuously innovate and advance our product offerings to simplify the ability to build AI-powered solutions on the Databricks Data Intelligence Platform safely. We believe there is no greater accelerant to delivering ML to production than building on a unified, data-centric AI platform. On Databricks, data and models can be managed and governed in a single governance solution with Unity Catalog. With Mosaic AI Model Serving, we streamlined the complexities associated with infrastructure for real-time model deployment, providing a scalable and user-friendly solution. For long-term efficiency and performance stability in ML production, Databricks Lakehouse Monitoring plays a pivotal role. This tool ensures continuous performance monitoring, contributing to sustained excellence in machine learning operations. These components collectively form the data pipelines of an ML solution, all of which can be orchestrated using Databricks Workflows.

Perhaps the most significant recent change in the machine learning landscape has been the rapid advancement of generative AI. Generative models such as large language models (LLMs) and image generation models have revolutionized the field, unlocking previously unattainable levels of natural language and image generation. However, their arrival also introduces new challenges and decisions to be made in the context of MLOps.

With all these developments in mind, below is a list of the necessary mitigation controls for organizations to address AI security risks. This mitigation guidance incorporates new Databricks features such as Models in Unity Catalog, Model Serving, and Lakehouse Monitoring into our MLOps architecture recommendations.

| CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON 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 RESPONSE 10.3, MODEL SERVING — INFERENCE RESPONSE 10.4

**DESCRIPTION**

Implementing single sign-on with an identity provider's (IdP) multi-factor 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 RESPONSE 10.3, MODEL SERVING — INFERENCE RESPONSE 10.4

**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

## 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 RESPONSE 10.3, MODEL SERVING — INFERENCE RESPONSE 10.4

**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

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

## 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 RESPONSE 10.3, MODEL SERVING — INFERENCE RESPONSE 10.4

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

## DASF 5 Control access to data and other objects

RISKS

RAW DATA 1.1, RAW DATA 1.4, 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 RESPONSE 10.3, MODEL SERVING — INFERENCE RESPONSE 10.4

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

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

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

## DASF 7 Enforce data quality checks on batch and streaming datasets

**RISKS**

RAW DATA 1.3 · RAW DATA 1.9 · 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

## DASF 8 Encrypt data at rest

**RISKS**

RAW DATA 1.4 · DATASETS 3.2 · DATASETS 3.3

**DESCRIPTION**

Databricks supports customer-managed encryption keys to strengthen data at rest protection and greater access control.

**CONTROL CATEGORY**

Configuration

**PRODUCT REFERENCE**

AWS | Azure | GCP

## DASF 9 Encrypt data in transit

**RISKS**

RAW DATA 1.4 · DATASETS 3.2 · DATASETS 3.3

**DESCRIPTION**

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.

**CONTROL CATEGORY**

Out-of-the-box

**PRODUCT REFERENCE**

AWS | Azure | GCP

## DASF 10 Version data

**RISKS**

RAW DATA 1.5 · RAW DATA 1.7

**DESCRIPTION**

Store data in a lakehouse architecture using Delta tables. Delta tables can be versioned to revert any user's or malicious actor's poisoning of data. Data can be stored in a 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 UC 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.

**CONTROL CATEGORY**

Implementation

**PRODUCT REFERENCE**

AWS | Azure | GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

## DASF 11 Capture and view data lineage

RISKS

RAW DATA 1.6 DATA PREP 2.1 DATASETS 3.1 GOVERNANCE 4.1 EVALUATION 6.1

DESCRIPTION

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.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 12 Delete records from datasets

RISKS

RAW DATA 1.8

DESCRIPTION

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.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 13 Use near real-time data

RISKS

RAW DATA 1.9

DESCRIPTION

Use Databricks for near real-time data ingestion, processing, machine learning, and AI for streaming data.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 14 Audit actions performed on datasets

RISKS

RAW DATA 1.10 DATASETS 3.1

DESCRIPTION

Databricks auditing, enhanced by Unity Catalog's 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.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | AWS | GCP

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

## 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<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| GCP |

## DASF 16 Secure model features

| RISKS | DESCRIPTION | CONTROL CATEGORY |
|---|---|---|
| DATA PREP 2.1<br>DATA PREP 2.2<br>DATASETS 3.1<br>GOVERNANCE 4.1<br>ALGORITHMS 5.2<br>MODEL SERVING — INFERENCE REQUESTS 9.10 | 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. | Implementation<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| GCP |

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

| RISKS | DESCRIPTION | CONTROL CATEGORY |
|---|---|---|
| DATA PREP 2.4<br>DATASETS 3.1<br>GOVERNANCE 4.1<br>ALGORITHMS 5.2 | 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. | Configuration<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| GCP |

## DASF 18 Govern model assets

| RISKS | DESCRIPTION | CONTROL CATEGORY |
|---|---|---|
| GOVERNANCE 4.1 | 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. | Configuration<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| GCP |

databricks



CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

## DASF 19 Manage end-to-end machine learning lifecycle

RISKS

GOVERNANCE 4.2 MODEL 7.1

DESCRIPTION

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.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 20 Track ML training runs

RISKS

ALGORITHMS 5.1 ALGORITHMS 5.3

DESCRIPTION

MLflow tracking facilitates the automated recording and retrieval of experiment details, including algorithms, code, datasets, parameters, configurations, signatures and artifacts.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

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

RISKS

RAW DATA 1.3 GOVERNANCE 4.2 ALGORITHMS 5.2

DESCRIPTION

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.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | N/A

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

RISKS

EVALUATION 6.2 MODEL 7.3

DESCRIPTION

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.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON 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 | GCP

## 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

## 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

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

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

## 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 RESPONSE 10.2
MODEL SERVING — INFERENCE RESPONSE 10.3
MODEL SERVING — INFERENCE RESPONSE 10.4
MODEL SERVING — INFERENCE RESPONSE 10.5

DESCRIPTION

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 | N/A

## 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

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 | N/A

## DASF 33 Manage credentials securely

RISKS

MODEL 7.2
MODEL MANAGEMENT 8.2

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

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

## DASF 34 Run models in multiple layers of isolation

| RISKS | DESCRIPTION | CONTROL CATEGORY |
|---|---|---|
| MODEL 7.1<br>MODEL SERVING — INFERENCE REQUESTS 9.3 | 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. | Out-of-the-box<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| N/A |

## DASF 35 Track model performance

| RISKS | DESCRIPTION | CONTROL CATEGORY |
|---|---|---|
| MODEL SERVING — INFERENCE RESPONSE 10.1 | 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. | Implementation<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| N/A |

## DASF 36 Set up monitoring alerts

| RISKS | DESCRIPTION | CONTROL CATEGORY |
|---|---|---|
| RAW DATA 1.3<br>MODEL SERVING — INFERENCE RESPONSE 10.1 | Databricks SQL alerts can monitor the metrics table for security-based conditions, ensuring data integrity and timely response to potential issues:<br>▪ **Statistic range Alert:** Triggers when a specific statistic, such as the fraction of missing values, exceeds a predetermined threshold<br>▪ **Data distribution shift alert:** Activates upon shifts in data distribution, as indicated by the drift metrics table<br>▪ **Baseline divergence alert:** Alerts if data significantly diverges from a baseline, suggesting potential needs for data analysis or model retraining, particularly in InferenceLog analysis | Implementation<br><br>PRODUCT REFERENCE<br>AWS \| Azure \| N/A |

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

## DASF 37 Set up inference tables for monitoring and debugging models

**RISKS**

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

**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 dataidentification:** Compiling data that needs relabeling

**CONTROL CATEGORY**

Implementation

**PRODUCT REFERENCE**

AWS | Azure | N/A

## DASF 38 Platform security — vulnerability management

**RISKS**

PLATFORM 12.1

**DESCRIPTION**

Managing vulnerabilities entails addressing complex security challenges with performance impact considerations. Databricks' formal and documented vulnerability management program, overseen by the chief security officer (CSO), is approved by management, undergoes annual reviews and is communicated to all relevant internal parties. The policy requires that vulnerabilities be addressed based on severity: critical vulnerabilities within 14 days, high severity within 30 days and medium severity within 60 days.

**CONTROL CATEGORY**

Out-of-the-box

**PRODUCT REFERENCE**

AWS | Azure | GCP

## DASF 39 Platform security — Incident Response Team

**RISKS**

PLATFORM 12.2 PLATFORM 12.3

**DESCRIPTION**

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.

**CONTROL CATEGORY**

Out-of-the-box

**PRODUCT REFERENCE**

AWS | Azure | GCP

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

## DASF 40 Platform security — internal access

RISKS

PLATFORM 12.4

DESCRIPTION

Databricks personnel, by default, do not 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 multi-factor authentication, and employees must connect to the Databricks VPN.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 41 Platform security — secure SDLC

RISKS

PLATFORM 12.5

DESCRIPTION

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.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 42 Employ data-centric MLOps and LLMOps

RISKS

DATA PREP 2.2, DATA PREP 2.3, DATA PREP 2.4, GOVERNANCE 4.2, ALGORITHMS 5.1, ALGORITHMS 5.3, EVALUATION 6.1, 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

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

## 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

## DASF 45 Evaluate models

**RISKS**

EVALUATION 6.1 · EVALUATION 6.2 · 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

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

## 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 multi-modal 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 Search is logically isolated, authenticated and authorized
- Mosaic AI Vector Search encrypts all data at rest (AES-256) and in transit (TLS 1.2+)

**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

## 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

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

## 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 | GCP

## 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

## 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

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

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

## DASF 54 Implement LLM 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

**DESCRIPTION**

Databricks supports guardrails to wrap around LLMs and help enforce appropriate behavior. Guardrails in Model Serving Foundation Model APIs can safeguard model inputs and outputs. Any content that is detected in the following categories is determined as unsafe:

- Violence and hate
- Sexual content
- Criminal planning
- Guns and illegal weapons
- Regulated or controlled substances
- Suicide and self-harm

To filter on other categories for custom pre- and post-processing. For example, to filter data that your company considers sensitive, such as PII/PHI from model inputs and outputs, wrap any regex or function and deploy it as an endpoint using Feature Serving.

**CONTROL CATEGORY**

Implementation

**PRODUCT REFERENCE**

AWS | Azure | N/A

## 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 RESPONSE 10.1

**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 starting point 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**

Private Preview
(Ask your account team)

CONTROL/RISK | DESCRIPTION OF CONTROL IMPLEMENTATION ON DATABRICKS PLATFORM

## DASF 57 Use attribute-based access controls (ABAC)

RISKS

MODEL SERVING — INFERENCE REQUESTS 9.10

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

## DASF 58 Protect data with filters and masking

RISKS

MODEL SERVING — INFERENCE REQUESTS 9.10

DESCRIPTION

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.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

## DASF 59 Use clean rooms

RISKS

RAW DATA 1.1 | RAW DATA 1.7 | MODEL MANAGEMENT 8.2 | MODEL SERVING — INFERENCE REQUESTS 9.10

DESCRIPTION

Building AI applications today necessitates collaborative efforts across organizations and teams, emphasizing a commitment to privacy and data security. Databricks Clean Rooms offer 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 ensure multiparty collaboration without compromising data privacy or security and enables organizations to build scalable AI applications in a privacy-safe manner.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS | Azure | GCP

databricks



05

# Conclusion

In an era defined by data-driven decision-making and intelligent automation, the importance of AI security cannot be overstated. The Databricks AI Security Framework provides essential guidance for securely developing, deploying and maintaining AI models at scale — and ensuring they 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 teams and security teams must actively collaborate to pursue their common goal of improving the security of AI systems. Whether you are implementing traditional machine learning solutions or LLM-driven applications, the core tenets of machine learning adoption remain constant:

## Databricks AI Security Framework (DASF)

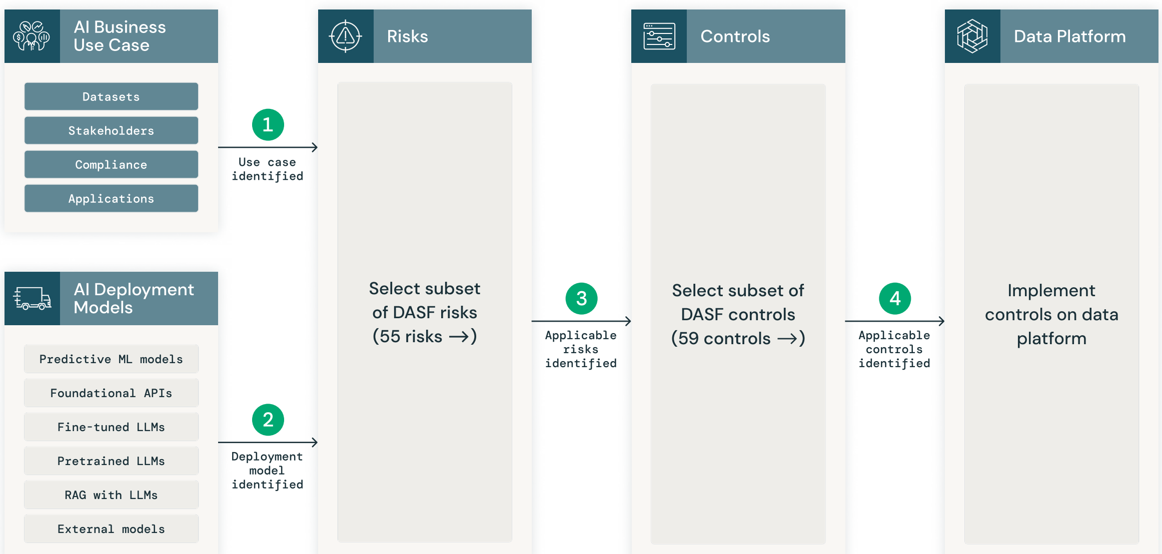



**Figure 2:** Implementation guidance of DASF controls on the Databricks Data Intelligence Platform.

While manually checking DASF controls is important when first configuring Databricks, we have 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 →

1. **Identify the AI business use case:** Always keep your business goals in mind. Make sure there is a well-defined use case with stakeholders you are trying to secure adequately, whether already implemented or in planning phases. This will help inform which AI system components are of greatest business value for any given business use case

2. **Determine the AI deployment model:** Choose an appropriate 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) to determine how shared responsibilities (especially for securing each component) are split across the 12 ML/GenAI components between your organization, the Databricks Data Intelligence Platform and any partners involved.

3. **Select the most pertinent risks:** From our documented list of 55 security risks, pinpoint the ones most relevant to your organization based on the outcome of step #2. Identify the specific threats linked to each risk and the targeted ML/GenAI component for every threat.

4. **Choose and implement controls:** Select controls 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.

Databricks 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. Red teaming and testing can help iteratively improve and mitigate discovered weaknesses of models. 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.

A lot of care has been taken to make this whitepaper accurate; however, as AI is an evolving field, please reach out to us if you have any feedback. If you're interested in participating in one of our AI Security workshops, please contact dasf@databricks.com.
If you are curious about how Databricks approaches security, please visit our Security and Trust Center.

# Resources and Further Reading

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

## AI and Machine Learning on Databricks

Training Course: Generative AI Fundamentals →
Webpage: AI and Machine Learning on Databricks →
Industry Solutions: Solution Accelerators →
Blogs: Responsible AI → | AI/ML Blogs →
eBooks: State of Data + AI → | Big Book of MLOps: 2nd Edition →
Learning Library: Generative AI Engineering With Databricks →

## Databricks Unity Catalog

Webpages: Databricks Unity Catalog → | AI Governance →
eBook: Data and AI Governance →

## 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

## Responsible AI on the Databricks Data Intelligence Platform

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

## Data Sharing and Collaboration

Webpage: Delta Sharing →

eBook: Data Sharing and Collaboration With Delta Sharing →

Blogs: What's New in Data Sharing and Collaboration → | AI Model Sharing →

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

## Industry Resource

An Architectural Risk Analysis of Machine Learning Systems →

NIST AI Risk Management Framework →

MITRE ATLAS Adversarial ML →

OWASP Top 10 for LLMs →

Guidelines for Secure AI System Development →

Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence →

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 →

## Third-Party Tools

**Model scanners:** HiddenLayer Model Scanner → | fickling → | ModelScan → | AI Risk Database → | NB Defense →

**Model validation tools:** Robust Intelligence continuous validation → | Vigil LLM security scanner → | Garak automated scanning → | HiddenLayer MLDR → | Citadel Lens →

**Guardrails for LLMs:** NeMo Guardrails → | Guradrails AI → | Lakera Guard → | Robust Intelligence AI Firewall → | Protect AI Guardian → | Arthur Shield → | Laiyer LLM Guard → | Amazon Guardrails → | Meta Llama Guard → | HiddenLayer AISec Platform →

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.

# 07 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 Officer


**Omar Khawaja**
Vice President, Field CISO


**Arun Pamulapati**
Senior Staff Security Field Engineer


**David Wells**
Staff Security Field Engineer


**Kelly Albano**
Product Marketing Manager


**Erika Ehrli**
Senior Director Product Marketing


**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


**Jeffrey Hirschey**
Senior Product Counsel


**Aliaksandra Nita**
Senior Technical Program Manager


**Neil Archibald**
Senior Staff Security Engineer


**Silvio Fiorito**
Principal Security Field Engineer


**Andrew Weaver**
Principal Specialist Solutions Architect

## CARNEGIE MELLON UNIVERSITY

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

## CAPITAL ONE FINANCIAL

**Ebrima N. Ceesay, PhD, CISSP**
Senior Distinguished Engineer

## 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

## PROTECT AI

**Diana Kelley**
CISO

## BARRACUDA

**Grizel Lopez**
Senior Director of Engineering

## 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

## META

**Brandon Sloane**
Risk Lead

# Appendix: Glossary

## A

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

**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.

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

## 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

**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 (or 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

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

**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 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.

**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.

**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.

**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

**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.

**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

**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.

**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.

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

**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.

**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 (e.g., the source class) to another (e.g., 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.

**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. Use 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 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 Model APIs. Developing a model requires a series of experiments and a way to track and compare the conditions and results of those experiments.

**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.

**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 Meta Llama 3.1, 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.

## N

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

## O

**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.

**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.

## 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.

**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

**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

**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."

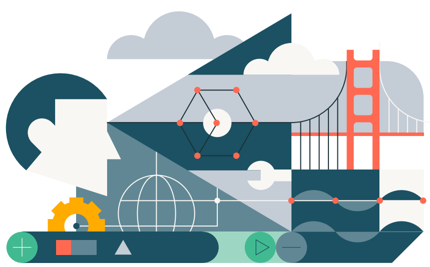
## W

**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.

# License





## About Databricks

Databricks is the data and AI company

Learn more →

## Security & Trust Center

Your data security is our priority

Learn more →

Databricks is the data and AI company. More than 10,000 organizations worldwide — including Block, Comcast, Condé Nast, Rivian, Shell and over 60% of the Fortune 500 — rely on the Databricks Data Intelligence Platform to take control of their data and put it to work with AI.

Databricks is headquartered in San Francisco, with offices around the globe, and was founded by the original creators of Lakehouse, Apache Spark™, Delta Lake and MLflow.

To learn more, follow Databricks on LinkedIn, X and Facebook.

Evaluate Databricks for yourself. Visit us at databricks.com and try Databricks free!