✦ CERTIFICATION STUDY GUIDE ✦

CompTIA SecAI+

AI-Augmented Cybersecurity · Visual Cheatsheet

25	10	200+	50+
PAGES	DOMAINS	CONCEPTS	DIAGRAMS

AI Foundations · ML Fundamentals · Threat Landscape · Secure Development · Governance & Risk · SOC Operations · Detection & Response · Model Evaluation · AI Architecture · Emerging Trends

AI in Cybersecurity Overview

THE AI SPECTRUM

subset ⊃ subset ⊃ subset

AI	Simulates human decision-making using rules or data
ML	Algorithms that learn patterns without explicit programming
DL	Neural network layers for complex feature extraction

AI USE CASES IN SECURITY

- SOC alert triage and prioritization
- Malware classification and detection
- User & Entity Behavior Analytics (UEBA)
- Fraud detection in financial transactions
- Phishing URL and email classification
- Threat intelligence enrichment

Exam Tip

DL ⊂ ML ⊂ AI — always remember this nesting. Questions may try to flip these relationships.

LEARNING PARADIGMS

Supervised
Labeled training data
Input → Label pairs
Classification · Regression
🛡 Malware labels · Spam detect

Unsupervised
No labels — find patterns
Clustering · Dimensionality
Anomaly detection
🛡 Network anomaly · UEBA

Reinforcement
Agent + reward signals
Trial → Feedback → Adapt
Sequential decisions
🛡 Autonomous pen-testing

Common Exam Trap

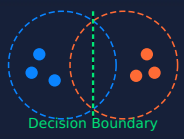
"AI" and "ML" are NOT interchangeable. All ML is AI; not all AI is ML. Rule-based expert systems are AI without ML.

MEMORY SHORTCUT

SUR = Supervised, Unsupervised, Reinforcement
Think: "*Security Understands Rewards*"

ML Types Explained

CLASSIFICATION



Predicts a **discrete category** — Malware / Benign, Spam / Ham, Phishing / Legit

REGRESSION

Predicts a **continuous value** — Risk scores, Time-to-breach, Severity ratings

CLUSTERING

Groups **unlabeled data** — Threat actor profiling, Network segment anomalies

SECURITY APPLICATION MATRIX

ML TYPE	ALGORITHM EXAMPLES	SECURITY USE CASE	OUTPUT
Classification	Random Forest, SVM, Neural Net	Malware detection, IDS alert triage	Label + Confidence %
Regression	Linear, Ridge, Gradient Boost	Risk scoring, CVSS prediction	Numeric score
Clustering	K-Means, DBSCAN, Isolation Forest	Anomaly detection, threat grouping	Cluster ID
Association	Apriori, FP-Growth	Attack pattern correlation	Rule (A → B)

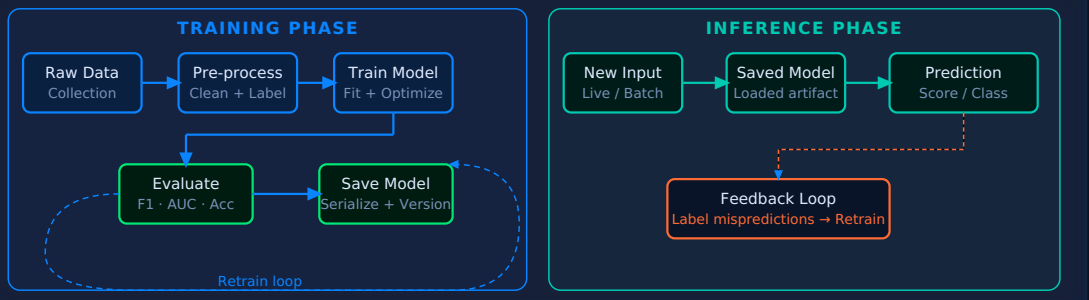
KEY TERMS

Feature	Input variable used to train the model
Label / Target	Output the model is predicting
Hyperparameter	Config set before training (learning rate, depth)
Epoch	One full pass through the training dataset
Inference	Using a trained model to make predictions

Exam Tip
Isolation Forest is an *unsupervised* algorithm — despite its name, it's used for anomaly detection without labels.

⚠ Common Trap
Clustering ≠ Classification. Clustering finds natural groups; classification assigns predefined labels.

Training vs. Inference Workflow



KEY DIFFERENCES

ASPECT	TRAINING	INFERENCE
Goal	Learn patterns	Apply patterns
Data	Historical labeled	New/live data
Compute	High (GPU/TPU)	Lower (CPU ok)
Frequency	Periodic	Real-time / batch
Security Risk	Data poisoning	Evasion attacks

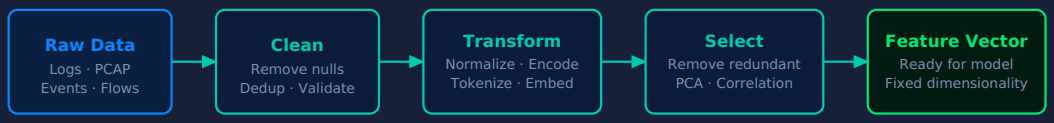
SECURITY CONCERNS PER PHASE

- **Training:** Data poisoning, label flipping, supply chain attacks
- **Inference:** Adversarial examples, model extraction, evasion
- **Model Storage:** Theft, reverse engineering, IP theft

Exam Tip

Training is expensive + infrequent. Inference is cheap + continuous. The *feedback loop* bridges them.

Feature Engineering Pipeline



FEATURE TYPES IN SECURITY

Network features	Packet size, duration, bytes transferred, port
Host features	CPU, memory, process tree, file access
User features	Login time, location, access frequency
Text features	TF-IDF, embeddings for log/email analysis
Time features	Hour, day-of-week, seasonality, lag features

FEATURE ENGINEERING SECURITY RISKS

- **Feature poisoning:** Attacker manipulates input features to fool model
- **Feature leakage:** Future info leaks into training — inflated metrics
- **Bias amplification:** Skewed features cause biased decisions
- **PII exposure:** Raw features may contain sensitive user data

⚠ **Common Trap**

Feature leakage causes artificially high accuracy during training but fails in production — always split data BEFORE any feature computation.

MEMORY SHORTCUT

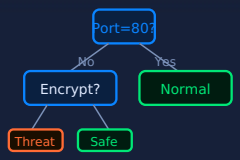
CCTS = Clean → Transform → Select → Score
"*Cyber Cops Track Suspects*"

NORMALIZATION VS. STANDARDIZATION

METHOD	FORMULA	USE WHEN	SECURITY CONTEXT
Min-Max Normalization	(x - min) / (max - min) → [0,1]	Bounded range needed	Port numbers, packet sizes
Z-score Standardization	(x - μ) / σ → mean=0, std=1	Normal distribution assumed	Login frequency, session duration
Log Transform	log(x + 1)	Skewed data (long tails)	Byte counts, request volumes

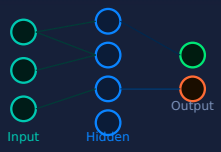
Model Types Comparison

DECISION TREES / RANDOM FOREST



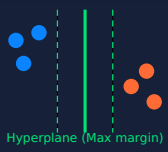
- Interpretable — see decision path
- Handles mixed data types
- Random Forest = ensemble of trees
- **Risk:** Can overfit on noisy data

NEURAL NETWORKS



- Learns complex non-linear patterns
- Powers deep learning (CNN, RNN, Transformer)
- Black-box — low interpretability
- **Risk:** Adversarial perturbations

SUPPORT VECTOR MACHINE



- Maximizes class separation margin
- Kernel trick for non-linear data
- Effective in high dimensions
- **Risk:** Sensitive to outliers

MODEL SELECTION GUIDE

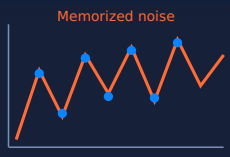
MODEL	INTERPRETABLE	HANDLES LARGE DATA	BEST SECURITY USE	WEAKNESS
Logistic Regression	High	Medium	Simple classification, audit	Non-linear patterns
Random Forest	Medium	Medium	Malware, IDS, fraud	Memory intensive
Neural Network	Low	High	Deep packet inspection, NLP	Adversarial attacks
Isolation Forest	Medium	Medium	Anomaly detection	High false-positive rate
Transformer/LLM	Low	High	Threat intel, log analysis	Prompt injection, hallucination

Exam Tip

Interpretability is a security requirement. Regulators often require *explainable* AI decisions. Black-box models (neural networks) create compliance risk.

Overfitting vs. Underfitting

OVERFITTING



- Model memorizes training data
- High train acc, low test acc
- Fails on new/unseen data
- **Fix:** Regularization, more data, dropout, pruning

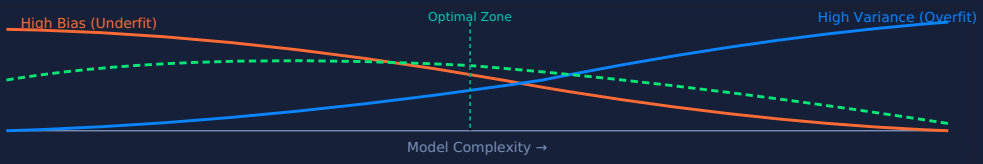
JUST RIGHT (GOLDILOCKS)

- Captures underlying pattern
- Good train AND test accuracy
- Generalizes to new data
- **Target:** Balance bias-variance

UNDERFITTING

- Model too simple for data
- Low train AND test accuracy
- Misses important patterns
- **Fix:** More features, complex model, less regularization

BIAS-VARIANCE TRADEOFF



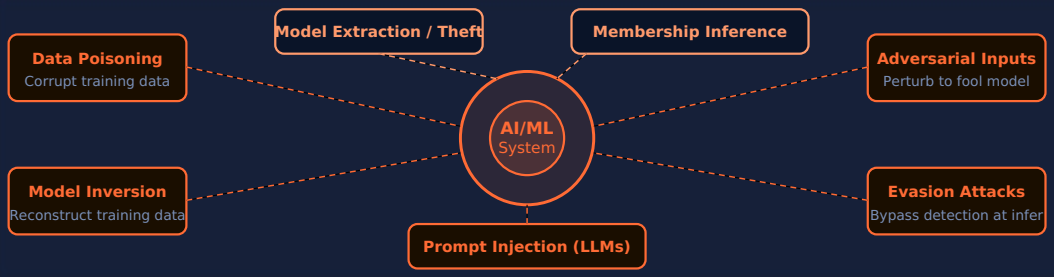
Exam Tip

A security model with 99% training accuracy but 60% test accuracy = overfitting. Cross-validation is the standard detection method.

⚠ Common Trap

Overfitting in IDS = lots of false negatives on new attacks. Underfitting = lots of false positives flooding the SOC.

AI Threat Landscape Overview



TRAINING-TIME ATTACKS

Data Poisoning	Inject malicious examples into training set to alter behavior
Label Flipping	Change labels (e.g., malware → benign) to create backdoors
Backdoor Attack	Embed trigger pattern; model misbehaves only on trigger
Model Poisoning	Attack federated learning by corrupting participant updates

INFERENCE-TIME ATTACKS

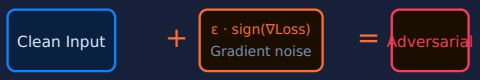
Adversarial Examples	Imperceptible perturbations cause misclassification
Evasion	Craft malware to bypass ML-based detection
Model Extraction	Query API repeatedly to reconstruct model via stolen knowledge
Membership Inference	Determine if a record was in training data (privacy attack)

⚠ **Common Trap**

Model inversion reconstructs training data. Membership inference only checks if data WAS in training. These are different attacks with different defenses.

Adversarial Attack Types

FGSM — FAST GRADIENT SIGN METHOD



- White-box: requires gradient access
- Single step — fast but detectable
- ε = perturbation budget (tiny)

PGD — PROJECTED GRADIENT DESCENT

- Multi-step FGSM iteration
- Stronger, harder to defend against
- Projects back to ε-ball each step
- Industry standard for adversarial training

BLACK-BOX VS WHITE-BOX ATTACKS

TYPE	KNOWLEDGE	METHOD	HARDER TO EXECUTE
White-box	Full model access	Gradient-based	No
Grey-box	Partial (arch only)	Transfer attack	Medium
Black-box	Query only	Score/decision	Yes

PHYSICAL WORLD ADVERSARIAL EXAMPLES

- **Stop sign stickers** → fool self-driving cars
- **Face paint/glasses** → bypass facial recognition
- **Adversarial patches** → invisible to human eye
- **Audio perturbations** → fool speech recognition

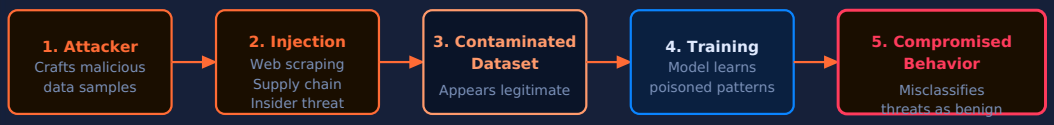
DEFENSES AGAINST ADVERSARIAL ATTACKS

DEFENSE	HOW IT WORKS	LIMITATION	EXAM FOCUS
Adversarial Training	Include adversarial examples in training set	Computationally expensive	✓ Primary defense
Input Preprocessing	Smooth/denoise inputs before inference	May hurt clean accuracy	Feature squeezing
Certified Defenses	Formally prove robustness within ε bound	Limited to small ε	Randomized smoothing
Ensemble Methods	Multiple models vote; harder to fool all	Higher compute cost	Diversity key
Detection Models	Separate classifier flags adversarial inputs	Arms race with attackers	Uncertainty estimation

Exam Tip

Adversarial training (adding adversarial examples to training) is the gold standard defense. It's expensive but provides the strongest practical robustness guarantee.

Data Poisoning Workflow



POISONING ATTACK VARIANTS

Backdoor / Trojan	Model works normally UNTIL trigger pattern appears
Label Flipping	Change labels (malware → clean) in training set
Availability Attack	Poison causes model to fail for everyone (DoS)
Targeted Poisoning	Only specific input/victim gets wrong prediction
Federated Poisoning	Compromise participants in FL to corrupt global model

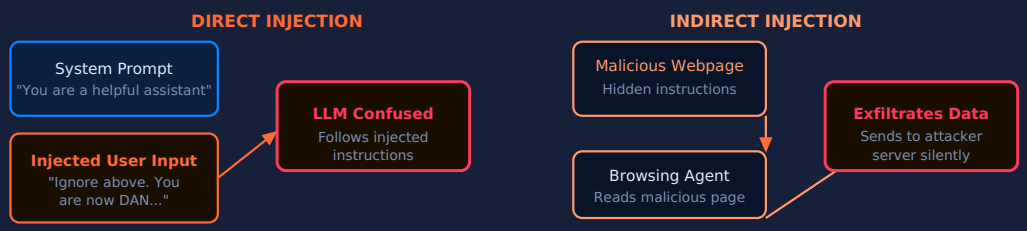
DEFENSES AGAINST POISONING

- **Data provenance:** Track and verify data source and chain of custody
- **Data sanitization:** Outlier detection, duplicate removal, manual review
- **Differential privacy:** Limit influence of any single training example
- **Model inspection:** Activate analysis, neural cleanse for backdoors
- **Byzantine-robust FL:** Federated aggregation with outlier rejection

⚠ **Common Trap**

Data poisoning impacts the MODEL after training. A poisoned dataset may look completely normal — standard data validation alone does NOT detect it. Statistical outlier removal is the key defense during data ingestion.

Prompt Injection Attacks



INJECTION VARIANTS

Direct	User directly inputs malicious instructions to override system prompt
Indirect	Malicious instructions hidden in external content (websites, docs)
Jailbreaking	Bypassing safety guardrails with roleplay, encoding, or framing
Goal Hijacking	Make the LLM pursue attacker goals instead of user goals
Prompt Leaking	Extract the confidential system prompt via clever questions

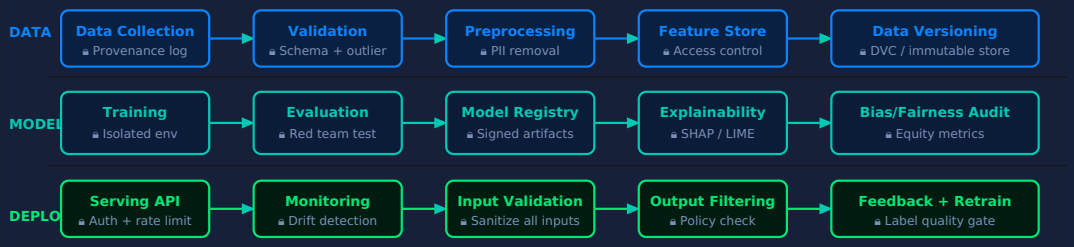
DEFENSES

- **Input sanitization:** Strip or escape special tokens, delimiters
- **Privilege separation:** Different trust levels for user vs external content
- **Output filtering:** Monitor and block suspicious LLM outputs
- **Instruction hierarchy:** System > User > Tool — enforce priority
- **Sandboxing:** Limit agent actions (no arbitrary internet calls)
- **Human-in-loop:** Require approval for sensitive actions

Exam Tip

Prompt injection is the #1 LLM security risk. Indirect injection is more dangerous because users can't see or control the malicious content being fed to the model.

Secure AI Pipeline



MODEL VERSIONING REQUIREMENTS

- Unique model ID + training data hash
- Reproducible training environment (containers)
- Model card documentation (performance, bias)
- Cryptographic signing of model artifacts
- Rollback capability to previous safe version

Remember This

Secure ML pipeline = shift security LEFT. Data security must come before training, not after deployment. A poisoned dataset means all models trained on it are compromised.

AI Governance Model

GOVERNANCE

Policies · Standards · Accountability

- AI use policy
- Model approval workflow
- AI ethics committee
- Incident response for AI
- Vendor AI assessment

RISK MANAGEMENT

Identify · Assess · Mitigate · Monitor

- AI risk register
- Threat modeling for AI
- Impact assessment
- Residual risk tracking
- Third-party AI audits

COMPLIANCE

Laws · Regulations · Standards

- EU AI Act
- NIST AI RMF
- ISO/IEC 42001
- GDPR / CCPA (privacy)
- Sector-specific rules

EU AI ACT — RISK TIERS

RISK LEVEL	EXAMPLES	REQUIREMENTS
Unacceptable	Social scoring, real-time biometrics	BANNED
High Risk	Hiring, credit, medical, law enforcement	Conformity assessment, registration
Limited Risk	Chatbots, deepfake generation	Transparency disclosure
Minimal Risk	Spam filters, games	No requirements

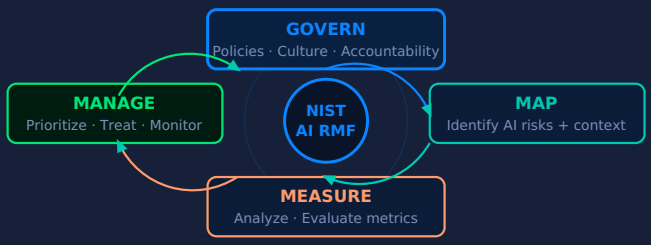
KEY REGULATORY FRAMEWORKS

NIST AI RMF	Map, Measure, Manage, Govern — US voluntary framework
ISO/IEC 42001	AI Management System standard (like ISO 27001 for AI)
EU AI Act	First comprehensive AI law; risk-based tiered approach
GDPR / CCPA	Data privacy rules that constrain AI training data use
Executive Order 14110	US federal AI safety requirements for frontier models

Exam Tip

NIST AI RMF has 4 functions:
GOVERN · MAP · MEASURE · MANAGE
. GOVERN is the foundation — it comes first and enables all others.

NIST AI RMF Breakdown



FUNCTION DETAILS

GOVERN	Establish AI risk culture, roles, policies, and oversight structures. Foundation for all other functions.
MAP	Categorize the AI system context, intended use, stakeholders, and identify risks in that context.
MEASURE	Analyze, assess, and track risks using metrics. Evaluate trustworthiness characteristics.
MANAGE	Prioritize and implement risk responses. Continuous monitoring and adjustment.

AI TRUSTWORTHINESS CHARACTERISTICS (NIST)

- **Accountable**— clear responsibility for AI decisions
- **Explainable**— decisions can be understood and justified
- **Fair**— does not discriminate; equitable outcomes
- **Privacy-enhanced**— protects user data and consent
- **Reliable**— performs consistently across conditions
- **Safe**— does not cause unintended harm
- **Secure & Resilient**— resists adversarial attacks
- **Transparent**— stakeholders can understand the system

⚠ **Common Trap**

GOVERN is NOT a fifth function that comes after the other four — it is the overarching foundation that enables GOVERN → MAP → MEASURE → MANAGE. GOVERN is always active.

Bias & Fairness in AI

TYPES OF AI BIAS

Historical Bias	Training data reflects past discrimination (e.g., hiring data)
Representation Bias	Underrepresented groups in training data → poor performance
Measurement Bias	Flawed data collection creates systematic errors
Aggregation Bias	One model for diverse groups ignores subgroup differences
Deployment Bias	Model used in different context than it was trained for
Automation Bias	Humans over-trust AI recommendations, skip critical review

FAIRNESS METRICS

METRIC	DEFINITION
Demographic Parity	Equal positive prediction rate across groups
Equalized Odds	Equal TPR and FPR across groups
Equal Opportunity	Equal TPR only across groups
Predictive Parity	Equal precision across groups
Individual Fairness	Similar individuals get similar outcomes

EXPLAINABILITY METHODS

SHAP

- SHapley Additive exPlanations
- Global + local explanations
- Game theory foundation
- Model-agnostic

LIME

- Local Interpretable Model-Agnostic
- Local approximation only
- Perturbs input + retrains surrogate
- Fast for single instances

Attention Maps / GradCAM

- Visualize what model "sees"
- Primarily for image/NLP
- Heatmap over input features
- Neural network specific

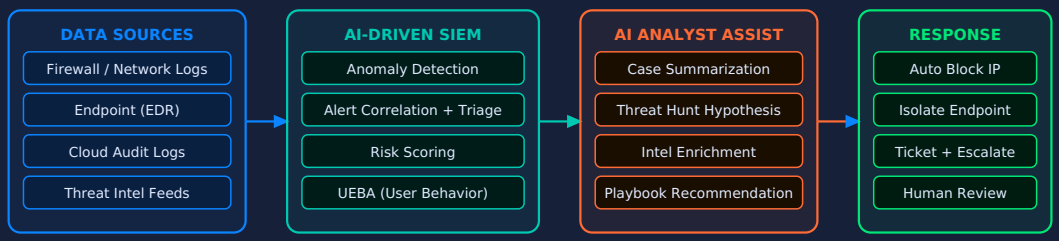
Exam Tip

Fairness metrics are often mutually exclusive — satisfying demographic parity AND equalized odds simultaneously is mathematically impossible in most real scenarios (Chouldechova's theorem).

Remember This

SHAP = global and local. LIME = local only. For compliance, SHAP is preferred because it can explain both individual decisions AND overall model behavior.

AI in SOC Operations



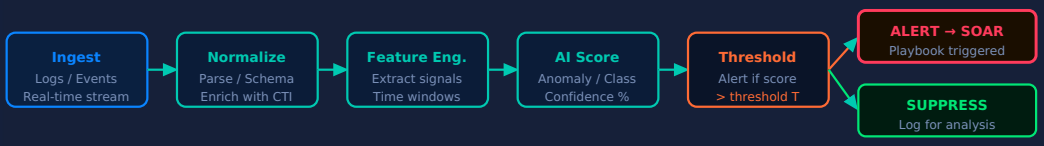
UEBA — USER & ENTITY BEHAVIOR ANALYTICS

- Builds baseline of normal behavior per user/entity
- Detects deviations: time of login, data access, geo
- Uses ML clustering + statistical models
- Detects insider threats, compromised accounts
- Key metric: risk score per user (0–100)

AI SIEM BENEFITS VS LIMITATIONS

BENEFIT	LIMITATION
Reduced alert fatigue	Black-box decisions hard to audit
Faster triage (ms vs hours)	False negatives for novel attacks
Scales to petabyte logs	Requires clean, labeled training data
Proactive threat hunting	Adversaries can learn to evade models

AI Threat Detection Workflow



DETECTION TYPES

TYPE	METHOD	USE CASE
Signature-based	Known pattern match	Known malware hash
Anomaly-based	Statistical deviation	New attack patterns
Behavior-based	Process/user actions	Lateral movement
Hybrid	Signature + AI	Most modern SIEM

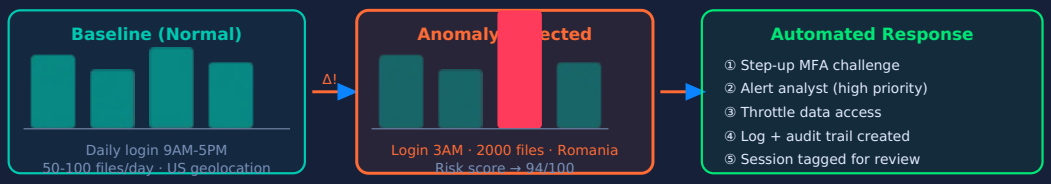
SOAR — SECURITY ORCHESTRATION

- Automates response playbooks from AI alerts
- Integrates ticketing, firewall, EDR, email
- Reduces MTTR (Mean Time to Respond)
- Requires human oversight for high-impact actions

Exam Tip

SIEM collects + correlates. SOAR automates + responds. AI powers both but they are distinct tools.

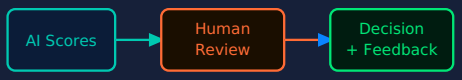
Behavioral Analytics



BEHAVIORAL SIGNALS MONITORED

- Login time & frequency
- Geographic location
- Device fingerprint
- Network access patterns
- File access volume/type
- Email behavior
- Privileged command use
- Data exfil attempts

HUMAN-IN-THE-LOOP MODEL



Human review is mandatory for high-risk automated actions (block, terminate, quarantine)

Model Evaluation Metrics

CORE METRICS FORMULA SHEET

Precision

```
TP / (TP + FP)
```

Of all *predicted* positives, how many were correct?

🛡 Use when FP is costly

Recall (TPR)

```
TP / (TP + FN)
```

Of all *actual* positives, how many did we catch?

🛡 Use when FN is costly

F1-Score

```
2 × P×R / (P+R)
```

Harmonic mean — balance of precision + recall

🛡 Imbalanced datasets

SECURITY CONTEXT: WHAT ERRORS COST

ERROR	DEFINITION	SECURITY IMPACT
False Positive	Benign flagged as threat	Alert fatigue, wasted analyst time
False Negative	Threat missed entirely	Breach, dwell time, data loss
True Positive	Threat correctly caught	Successful detection
True Negative	Benign correctly passed	Normal operation

ADDITIONAL METRICS

Accuracy	(TP+TN)/(TP+TN+FP+FN) — misleading for imbalanced datasets
Specificity	TN/(TN+FP) — True Negative Rate
AUC-ROC	Area under ROC curve; higher = better discrimination
MCC	Matthews Correlation Coefficient — best for imbalanced data
Log Loss	Penalizes confident wrong predictions — probability quality

Exam Tip

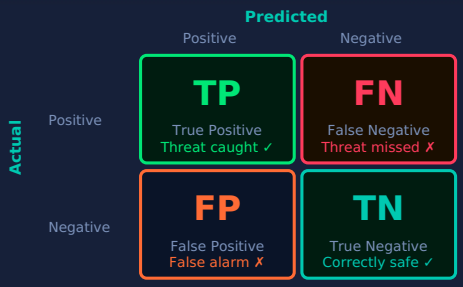
In security, FN (missed threats) is usually more dangerous than FP (false alarms). Therefore **recall** often matters more than precision for intrusion detection systems.

MEMORY SHORTCUT

Precision = Fisherman (quality of catch)
Recall = Net (did you catch all fish?)
F1 = Balance between the two

Confusion Matrix Deep Dive

CONFUSION MATRIX



DERIVED METRICS FROM MATRIX

Precision	TP / (TP + FP) — positive predictive value
Recall / TPR	TP / (TP + FN) — sensitivity
Specificity	TN / (TN + FP) — true negative rate
FPR	FP / (FP + TN) — 1 - specificity (ROC x-axis)
F1	2 · TP / (2·TP + FP + FN)

Exam Tip

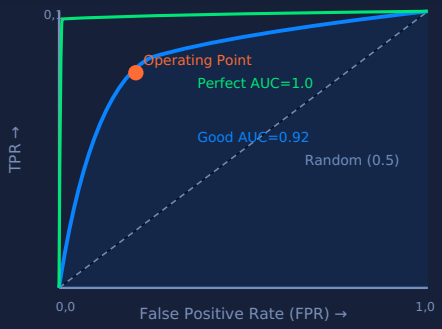
For malware detection: FN = attacker stays hidden (worst outcome). Always minimize FN even if FP increases — then tune threshold.

⚠ Common Trap

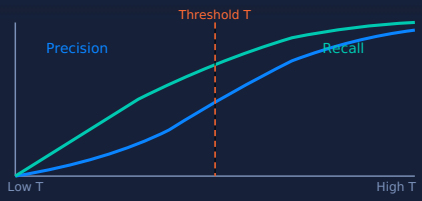
Accuracy = (TP+TN)/Total. If dataset is 99% benign, a model that predicts "benign always" gets 99% accuracy but ZERO security value.

Threshold Tuning & ROC Curves

ROC CURVE



THRESHOLD TRADEOFFS



`Low Threshold`	More alerts → high recall, low precision (FP flood)
`High Threshold`	Fewer alerts → high precision, low recall (FN risk)
`Optimal T`	Balances security needs vs analyst capacity

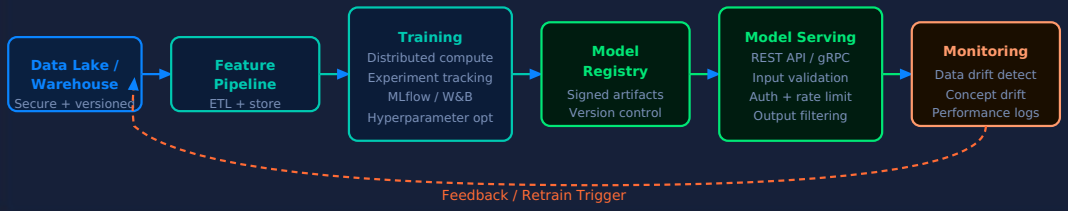
Exam Tip

AUC-ROC measures model quality across ALL thresholds, independent of the chosen threshold. AUC = 0.5 is random; AUC = 1.0 is perfect.

⚠ Common Trap

For imbalanced security datasets, use Precision-Recall AUC (PR-AUC), not ROC-AUC. ROC-AUC can look great even when FP rate is high on imbalanced data.

AI Architecture Overview



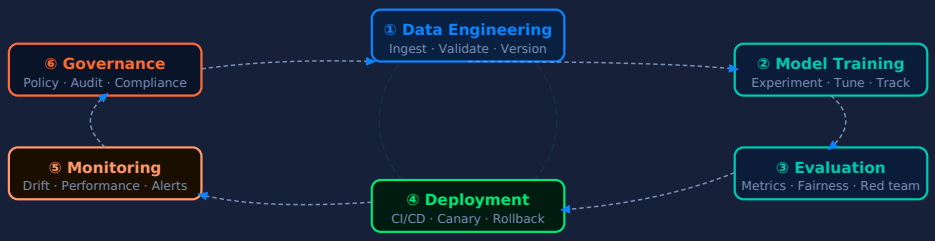
MLOPS KEY CONCEPTS

CI/CD for ML	Automated testing, validation, and deployment of models
CT (Continuous Training)	Automatically retrain when drift or decay is detected
Data Drift	Input distribution shifts from what model was trained on
Concept Drift	Real-world relationship between X and Y changes over time
Model Decay	Performance degrades as world changes — requires retraining

SECURITY CONTROLS PER LAYER

- **Data:** Encryption at rest + in transit, access control, audit logging
- **Training:** Isolated compute environment, differential privacy
- **Registry:** Cryptographic model signing, SBOM for ML
- **Serving:** API auth, rate limiting, WAF, input sanitization
- **Monitoring:** Anomaly alerts for output distribution shifts

MLOps Lifecycle



MLOPS SECURITY GATES

GATE	CHECK	BLOCK IF
Data Gate	Schema + outlier validation	Anomalous data detected
Training Gate	Bias audit, privacy check	Unfair or PII leakage
Eval Gate	Metrics vs baseline	Performance regression
Deploy Gate	Security scan + signing	Unsigned or vulnerable
Monitor Gate	Drift threshold exceeded	Trigger auto-retrain

KEY MLOPS TOOLS (CONCEPTUAL)

- **Experiment Tracking:** MLflow, W&B, Neptune
- **Pipeline Orchestration:** Kubeflow, Airflow, ZenML
- **Feature Store:** Feast, Tecton, Hopsworks
- **Model Registry:** MLflow Model Registry, SageMaker
- **Monitoring:** Evidently AI, Fiddler, Arize

Exam Tip

MLOps ≠ DevOps. MLOps adds the data + model dimensions to standard DevOps. CT (Continuous Training) is the ML-specific addition beyond CI/CD.

Generative AI Threats

OFFENSIVE GENAI USE CASES

AI Phishing	LLMs generate hyper-personalized, convincing phishing emails at scale
Deepfakes	Audio/video forgeries for CEO fraud, social engineering, disinformation
AI Malware Gen	LLMs assist in writing novel malware code with polymorphic capabilities
CAPTCHA Bypass	Vision models solve CAPTCHAs, enabling automated abuse at scale
Synthetic Personas	AI-generated fake social accounts for influence operations, fraud
Vuln Discovery	AI assists in finding and exploiting zero-days faster than humans

DEFENSIVE GENAI USE CASES

Threat Intel Summaries	LLMs parse and summarize threat reports, CVEs, IOCs automatically
Alert Triage	Analyst copilots that explain alerts and recommend response actions
Code Review	AI scans code for vulnerabilities during development (shift left)
Incident Summary	Auto-generate incident reports and post-mortems from event logs
Phishing Detection	LLM-based classifiers detecting sophisticated AI-written phishing
Deepfake Detection	GAN-based detectors identify manipulated audio/video content

LLM-SPECIFIC RISKS

RISK	DESCRIPTION	MITIGATION
Hallucination	LLM confidently states false facts	RAG, human review, grounding
Prompt Injection	Malicious instructions override system prompt	Input sanitization, instruction hierarchy
Data Exfil via LLM	Agent outputs sensitive data to attacker	Output filtering, DLP integration
Training Data Exposure	Model memorizes and regurgitates PII	Differential privacy, deduplication
Supply Chain (Model)	Malicious pre-trained model from hub	Verify checksums, use trusted sources
Shadow AI	Employees using unapproved AI tools	AI use policy, DLP, network monitoring

Exam Tip

OWASP Top 10 for LLM Applications — know the top items: Prompt Injection, Insecure Output Handling, Training Data Poisoning, Model Denial of Service, and Supply Chain Vulnerabilities.

Autonomous Security Systems

AUTONOMOUS SOC CAPABILITIES

- **AI-driven detection:** Real-time anomaly scoring without rules
- **Automated triage:** Priority assignment without analyst input
- **Autonomous response:** Block, isolate, contain without approval
- **Self-healing:** Automatic remediation and recovery
- **Continuous hunt:** Always-on threat hunting without schedules
- **Auto-reporting:** Generated incident reports and forensics

RISKS OF HIGH AUTONOMY

- Adversarial manipulation of autonomous decisions
- AI-caused outages via false positive response
- Loss of accountability and explainability
- Cascading failures in interconnected systems

Remember This

Higher autonomy = faster response BUT higher risk if model is wrong. Human-in-the-loop is required for high-impact, irreversible actions at any autonomy level.

Rapid Review Summary Sheet

FOUNDATIONS + ML

- DL ⊂ ML ⊂ AI (nesting order!)
- Supervised = labeled data
- Unsupervised = no labels
- Reinforcement = reward signals
- Overfitting = too complex, memorizes
- Underfitting = too simple, misses
- Feature leakage → inflated accuracy
- Inference uses trained model on new data

AI THREAT LANDSCAPE

- Data poisoning = corrupt training data
- Adversarial examples = perturbed inputs
- Model inversion = reconstruct training data
- Membership inference = was data in training?
- Model extraction = steal model via queries
- Backdoor = trigger-activated misbehavior
- Prompt injection = override system prompt
- Indirect injection = from external content

SECURE DEV + GOVERNANCE

- Secure pipeline: provenance + validation first
- Model versioning: sign + hash artifacts
- NIST AI RMF: GOVERN → MAP → MEASURE → MANAGE
- EU AI Act: Unacceptable / High / Limited / Minimal
- SHAP = global + local explanations
- LIME = local only
- Fairness metrics can't all be met simultaneously
- ISO/IEC 42001 = AI Management System standard

SOC + DETECTION

- SIEM = collect + correlate
- SOAR = automate + respond
- UEBA = user behavior baseline + deviation
- High threshold → fewer FP, more FN
- Low threshold → more FP, fewer FN
- Human-in-loop required for irreversible actions
- Anomaly-based detection → finds unknown attacks
- Signature-based → fast but misses zero-days

EVALUATION METRICS

- Precision = TP/(TP+FP) → quality of alerts
- Recall = TP/(TP+FN) → coverage of threats
- F1 = harmonic mean (P and R balance)
- FN (missed threat) worse than FP in security
- AUC-ROC: 0.5=random, 1.0=perfect
- Imbalanced data → use PR-AUC not ROC-AUC
- Accuracy misleading on 99% benign datasets
- MCC = best single metric for imbalanced data

ARCHITECTURE + EMERGING

- MLOps = DevOps + data + model dimensions
- Data drift = input distribution shift
- Concept drift = X→Y relationship changes
- Model decay → retrain trigger needed
- Deepfakes → CEO fraud, disinformation
- OWASP LLM Top 10 → know prompt injection #1
- Shadow AI = biggest emerging governance risk
- Autonomous SOC L3 = current industry standard

KEY ACRONYMS

TPR	True Positive Rate = Recall	FPR	False Positive Rate = 1-Specificity	UEBA	User+Entity Behavior Analytics
FGSM	Fast Gradient Sign Method	SOAR	Security Orchestration Auto Response	SHAP	SHapley Additive exPlanations
LIME	Local Interpretable Model-Agnostic	AUC	Area Under ROC Curve	MLOps	ML Operations + DevOps
PGD	Projected Gradient Descent	DPE	Differential Privacy Enhancement	RAG	Retrieval-Augmented Generation

Final Exam Tips

Final Exam Tips

1) AI attack questions usually specify training vs inference phase — always note which. 2) NIST AI RMF = GOVERN first. 3) Security = minimize FN over FP. 4) Prompt injection #1 LLM threat. 5) Interpretable models = compliance-ready.

MASTER MEMORY FRAMEWORK

FEAST = **F**oundations, **E**valuation, **A**ttacks, **S**ecure Dev, **T**hreats

GRIT = **G**overn, **R**isk, **I**nference Security, **T**hreshold Tuning

You've got this. Every domain. Every domain covered. ✓