CHECKLIST

Checklist: Validating AI Anomaly Detection for Network Logs

A practical vendor-neutral checklist for evaluating whether AI-based anomaly detection is ready for production in network log environments.

Checklist: Validating AI Anomaly Detection for Network Logs

Use this checklist to determine whether AI-based anomaly detection is a good fit for your logs, how to prepare the data, and what to verify before production use.

1) Define the detection goal

- [ ] Identify the exact behavior you want to surface:
- [ ] Beaconing
- [ ] Lateral movement
- [ ] Unusual egress destinations
- [ ] Service-to-service change
- [ ] Failed authentication bursts
- [ ] Other: _____
- [ ] Choose a consistent observation unit:
- [ ] Host
- [ ] Service account
- [ ] Subnet
- [ ] Container

- [ ] Session
- [ ] Destination pair
- [ ] Define what counts as a meaningful anomaly for this use case.
- [ ] Confirm the output will be used for prioritization, not automatic compromise declaration.

2) Confirm the environment is suitable

- [ ] The environment produces enough log volume to establish a stable baseline.
- [ ] Normal behavior is repeatable enough to compare over time.
- [ ] Logs cover the activity you want to detect.
- [ ] The target use case is not dominated by constant maintenance, automation, or churn.
- [ ] You have a reasonable historical window to learn baseline patterns.
- [ ] The data is not so sparse that the model will mostly learn noise.

3) Review log quality before modeling

- [ ] Fields are consistently named across sources.
- [ ] Timestamps are accurate and normalized.
- [ ] Source, destination, protocol, and identity fields are present where expected.
- [ ] Missing values are understood and handled consistently.
- [ ] Duplicates, dropped events, and parsing errors are measured.
- [ ] Log transformations do not destroy important behavioral signals.
- [ ] Normalization and enrichment happen before anomaly scoring.

4) Choose behavior-based features

- [ ] Use aggregated features rather than only raw events.

- [ ] Include time-windowed counts and rates.
- [ ] Include source and destination metadata.
- [ ] Include byte and packet volume features.
- [ ] Include protocol and port distribution signals.
- [ ] Include failure and reset rates.
- [ ] Include authentication-related burst patterns if relevant.
- [ ] Include sequence-based changes, such as a new destination after a privilege change.
- [ ] Verify the selected features reflect behavior over time.

5) Design the time window strategy

- [ ] Select a rolling window size that fits the behavior speed:
- [ ] 5 minutes
- [ ] 15 minutes
- [ ] 1 hour
- [ ] Other: _____
- [ ] Test whether the chosen window captures both short bursts and sustained deviations.
- [ ] Confirm windows compare like with like.
- [ ] Avoid mixing unrelated entities in the same baseline.

6) Establish a baseline

- [ ] Decide whether the approach is statistical, clustering-based, isolation-based, reconstruction-based, or hybrid.
- [ ] Calibrate the baseline on normal historical behavior.
- [ ] Separate routine variation from meaningful deviation.
- [ ] Verify the baseline reflects the current environment, not an outdated one.
- [ ] Check whether the log distribution has shifted over time.
- [ ] Recalibrate if traffic patterns have changed significantly.

7) Validate explainability and analyst usability

- [ ] Each alert includes a novelty score.
- [ ] Each alert includes the main contributing features.
- [ ] Analysts can compare the event to recent history.
- [ ] The model output is understandable without specialized data science tooling.
- [ ] The triage process explains why the window was flagged.
- [ ] The result can be defended during incident review.

8) Add operational context before alerting

- [ ] Asset criticality is available to the decision layer.
- [ ] Known maintenance windows are accounted for.
- [ ] Allowlisted automation is excluded or downweighted.
- [ ] Business-approved anomalies do not page the team unnecessarily.
- [ ] Detection output is routed through rules or policy before alerting.
- [ ] Decisioning is separated from scoring.

9) Test for false positives and false negatives

- [ ] Review a representative sample of flagged windows.
- [ ] Measure how often alerts are operationally useful.
- [ ] Check whether static thresholds would have performed better for some cases.
- [ ] Identify legitimate bursts that were incorrectly scored as suspicious.
- [ ] Identify stealthy changes the model missed.
- [ ] Validate on known benign and known suspicious patterns where possible.

10) Check production readiness

- [ ] A tuning process exists for thresholds and features.
- [ ] Recalibration cadence is defined.
- [ ] Monitoring for data drift is in place.
- [ ] Alert volume is within analyst capacity.
- [ ] Escalation criteria are documented.
- [ ] Logging and scoring failures are observable.
- [ ] Rollback or fallback behavior is defined.
- [ ] There is a plan to broaden or narrow scope after initial rollout.

11) Confirm scope is appropriately narrow

- [ ] Start with one high-value use case before expanding.
- [ ] Prefer stable, repeatable traffic patterns first.
- [ ] Avoid broad coverage if the environment changes too frequently.
- [ ] Expand only after the first deployment is producing actionable results.

12) Decide whether to proceed

Mark the deployment as ready only if all of the following are true:

- [ ] The logs are complete enough for the intended use case.
- [ ] The environment has a stable enough baseline.
- [ ] The model produces explainable outputs.
- [ ] The alert volume is manageable.
- [ ] Operational context is applied before paging analysts.
- [ ] The team has a plan to monitor drift and retrain or recalibrate.

Quick go/no-go summary

Go if:

- You have enough historical data
- The behavior is repeatable
- The features are well engineered
- Analysts can understand why alerts fired
- Context filtering reduces noise

No-go or not yet if:

- Logs are sparse or inconsistent
- Normal behavior changes constantly
- The model cannot explain itself
- False positives would overwhelm the team
- The use case is too broad for a first rollout

Suggested first deployment targets

- [ ] VPN authentication anomalies
- [ ] Internal lateral movement patterns
- [ ] Unusual egress destinations
- [ ] Service-to-service communication changes
- [ ] Failed login bursts from a single host

Notes

- AI is best used as a prioritization layer.
- A high anomaly score means unusual, not automatically malicious.

- The best production outcome combines normalization, rolling-window features, anomaly scoring, and contextual decisioning.

Worksheet:

- Primary use case: _____
- Entity type: _____
- Window size: _____
- Baseline period: _____
- Top features: _____
- Context filters: _____
- Alert owner: _____
- Review cadence: _____