Bounded Neural Authority: A Neuro-Symbolic Architecture for Detection and Response in Security Operations

Abstract

Modern Security Operations Centres (SOCs) rely heavily on automated detection pipelines. These pipelines take raw telemetry, normalize it, apply detection logic written by security teams, and turn the resulting findings into alerts for analysts to investigate. This approach works well because it is predictable and auditable, but it is also limited by what detection engineers can anticipate and encode in advance. Neural models offer a different set of strengths. They can identify patterns and relationships across large volumes of data that may be difficult to express as rules, but their lack of transparency and predictability makes it difficult to rely on them directly for production alerting.

This paper proposes a *neuro-symbolic detection and response* (NS-D&R) architecture that brings these two approaches together without replacing the existing detection pipeline. The rule-based pipeline remains the foundation, while neural models provide additional context through embeddings, anomaly detection, similarity search, graph analysis, causal reasoning, and large language model (LLM) assistance. The division of responsibility is deliberate: neural models are used to *discover, represent, rank, summarize, and propose hypotheses*, while symbolic components *validate, constrain, explain, deduplicate, score, and govern* their influence on production decisions. The architecture is organized into three planes—detection, reasoning, and response—and includes a detection knowledge graph and a bounded scoring mechanism designed to prevent neural outputs from dominating alert decisions. We also describe a phased adoption approach that retains human approval and auditability, together with an evaluation framework covering detection performance, neuro-symbolic behavior, and operational cost. The architecture is vendor-neutral and can be applied to any security detection stack built around normalized telemetry.

Keywords

neuro-symbolic AI, security operations, detection engineering, incident response, knowledge graphs, explainable AI

1. Introduction

Security detection engineering has converged on a common pipeline shape: telemetry is collected from endpoints, identity providers, cloud control planes, networks, and applications; it is normalized to a shared schema; detection analytics (queries or rules) generate structured findings; findings are enriched and scored; related findings are clustered into incidents; and a final alerting stage produces prioritized alerts that analysts triage and resolve. This pipeline is effective and, critically, governable: each detection is authored, versioned, and reviewed by humans, and each alert can be traced back to the logic that produced it.

The limitation of this model is that its expressive power is bounded by what detection engineers can explicitly encode. Subtle behavioral deviations, semantically similar variants of known attacks, and multi-entity incidents that unfold across users, devices, cloud accounts, and applications are difficult to capture with deterministic rules alone. Machine learning, particularly representation learning and sequence modeling, excels at exactly these tasks, but introduces opacity and instability that are unacceptable as the sole basis for production alerting in high-stakes environments.

We argue that the productive resolution is not to replace the symbolic pipeline with a neural one, nor to bolt on a parallel "AI alerting" system, but to construct a neuro-symbolic architecture in which neural components provide probabilistic context and hypotheses while symbolic components retain authority over validation, scoring bounds, explanation, and governance. This paper makes the following contributions:

- A generalized NS-D&R architecture organized as three planes, detection, reasoning, and response, that augments an existing signal-based pipeline without displacing it.
- A detection knowledge graph formulation that converts findings, evidence artifacts, and operational metadata into nodes and edges suitable for both symbolic queries and graph-based learning.

- A bounded confidence-calibration model that combines human-authored scores, symbolic rule deltas, and neural context under explicit limits, preventing neural models from dominating production decisions.
- A phased adoption roadmap and evaluation framework that prioritize auditability, safe fallback, and human approval throughout.

2. Background and Related Work

2.1 Neuro-Symbolic Artificial Intelligence

Neuro-symbolic artificial intelligence (NeSy AI) brings together two approaches to AI that have different strengths. Neural methods are particularly good at finding patterns and learning representations from large, noisy, and complex datasets. The difficulty is that what they learn can be hard to interpret, and their outputs are not always easy to constrain using explicit domain knowledge. Symbolic systems approach the problem from the other direction. They represent knowledge and rules explicitly, making them easier to inspect, modify, and use for deterministic or logically constrained reasoning. Their limitation is that much of this knowledge has to be defined in advance, making them less effective at discovering patterns that have not already been encoded (Besold et al., 2017; d'Avila Garcez et al., 2019; Garnelo & Shanahan, 2019; Hitzler et al., 2022).

Bringing the two together is therefore about more than combining two AI techniques. The aim is to give each approach the kind of work it is better suited to. Hitzler et al. (2022) identify several ways this can be done, including using symbolic knowledge to guide neural systems, extracting symbolic representations from learned models, using background knowledge to improve explainability, and combining neural and symbolic components for more complex reasoning tasks. d'Avila Garcez et al. (2019) make a similar case for neural-symbolic computing as a way of combining learning with explicit knowledge and reasoning, particularly in settings where interpretability and accountability matter.

There is no single technical model for doing this. Logic Tensor Networks (LTNs), for example, represent first-order logical constructs within differentiable neural computation,

allowing learned information and abstract knowledge to be considered together (Badreddine et al., 2022). DeepProbLog combines neural predicates with probabilistic logic programming, connecting learned perception with symbolic probabilistic inference (Manhaeve et al., 2018). NeurASP takes neural-network predictions and incorporates them into Answer Set Programming, where they can be evaluated alongside explicit symbolic rules and constraints (Yang et al., 2020). Embed2Sym approaches the problem differently by identifying symbolic concepts from clusters within a learned embedding space and then using those concepts as part of symbolic reasoning (Aspis et al., 2022).

These approaches integrate neural and symbolic reasoning within a computational framework. NS-D&R instead uses separate components that exchange information through defined interfaces. It does not propose a new neuro-symbolic learning algorithm. Instead, it considers how neural and symbolic capabilities can be combined within an operational detection-and-response system. Neural models are used where learning from data is useful—for representation, similarity, anomaly detection, generalization, and hypothesis generation—while symbolic mechanisms remain responsible for validation, explicit constraints, policy enforcement, scoring authority, and governance.

2.2 Neuro-Symbolic AI for Cyber Defence

Cyber defence is a natural setting for neuro-symbolic methods because security operations already use both neural and symbolic forms of reasoning. Anomaly detection, behavioural models, and embeddings can provide probabilistic evidence about suspicious activity, while SOCs rely on detection rules, signatures, schemas, adversary-technique mappings, allowlists, policies, playbooks, and response procedures. In many respects, the symbolic layer already exists. The challenge is combining it with learned models without giving up the predictability and control expected from a production security system.

Recent research has explored this combination across intrusion detection, alert contextualisation, threat hunting, and structured cyber knowledge (Eckhoff et al., 2025). Knowledge-enhanced anomaly detection provides another example. KnowGraph combines graph-based representation learning with logical reasoning and domain knowledge, including for enterprise intrusion detection (Zhou et al., 2024). Such approaches are relevant where training data cannot fully represent future adversary behaviour and established knowledge can constrain learned representations.

Existing work demonstrates the potential of neuro-symbolic methods, but largely addresses particular security tasks. A production SOC has a longer decision chain: telemetry is normalised, findings are generated, enriched and scored, related activity is grouped, alerts are created, and analysts investigate and respond. Introducing neural reasoning into this chain therefore raises broader questions about where a model should influence production

decisions, how far that influence should extend, how decisions can be reconstructed, and what happens when the model is unavailable or unreliable.

These questions matter because operational machine learning involves more than benchmark performance. Previous work has identified a gap between machine-learning research for network intrusion detection and the requirements of practical deployment (Apruzzese et al., 2023). Accuracy, precision, and recall remain important, but so do alert volume, analyst workload, latency, explainability, failure behaviour, and the consequences of incorrectly confident models.

2.3 Graph-Based Detection, Correlation, and Attack Reasoning

Graph-based security research shows how relationships between events, entities, and actions can reveal malicious activity that appears benign or inconclusive in isolation. Provenance and attack-graph approaches use this relational context to identify and reconstruct activity across time.

Tactical Provenance Graphs reason over causal relationships between EDR alerts and produce compact representations of multi-stage attacks, with threat scoring incorporating their temporal ordering (Hassan et al., 2020). KAIROS is a provenance-based intrusion detection system that uses graph neural networks to learn the temporal evolution of whole-system provenance graphs, identify anomalous system events, and reconstruct compact attack footprints for investigation (Cheng et al., 2024).

Together, these approaches demonstrate the value of relational context, temporal ordering, and graph learning for multi-stage attack analysis. NS-D&R builds on these ideas at a different level of abstraction, representing artifacts from across the detection-and-response lifecycle within a persistent graph. This allows the graph to serve both as a source of operational facts and as a structure from which learned models can propose additional relationships.

2.4 Confidence, Explainability, and Human Authority

Using neural outputs in operational security raises an important question: how much authority should be attached to model confidence? A confident prediction is not necessarily correct. Modern neural networks can be poorly calibrated, meaning confidence does not always

reflect the likelihood of a correct prediction (Guo et al., 2017). In a security pipeline, an incorrect model could suppress malicious activity or unnecessarily escalate benign activity.

Explainability presents a related problem. Machine-learning systems are easier to assess when they provide understandable evidence for their predictions (Ribeiro et al., 2016). For a SOC, however, explaining a recommendation does not mean the model should be allowed to act on it. An understandable recommendation may still be wrong, incomplete, or based on insufficient context.

NS-D&R therefore separates neural confidence from neural authority. A model may have high confidence in an anomaly, similarity, or attack hypothesis without authority to act on it independently. Neural outputs are treated as evidence evaluated by symbolic rules rather than as direct production commands. This distinction underpins the bounded confidence-calibration mechanism described later in the paper: neural context can influence risk within a defined range, but cannot independently suppress an otherwise alertable finding or assign the highest severity.

Human oversight provides a further boundary. The architecture applies different levels of automation according to the consequences of a decision. Neural assistance can operate with relatively few restrictions when retrieving similar incidents or summarising evidence, is more tightly controlled when changing risk scores or grouping findings, and requires human approval for consequential response actions. Authority is therefore determined by the potential consequence of a decision, not by model confidence.

2.5 Research Gap

The literature reviewed above establishes many of the capabilities needed for neuro-symbolic cyber defence. Neuro-symbolic research has explored combinations of learned representations, explicit knowledge, logical reasoning, and symbolic constraints (Aspis et al., 2022; Badreddine et al., 2022; Besold et al., 2017; d'Avila Garcez et al., 2019; Garnelo & Shanahan, 2019; Hitzler et al., 2022; Manhaeve et al., 2018; Yang et al., 2020). Cybersecurity research has applied these ideas to intrusion detection, alert contextualisation, anomaly detection, and threat hunting (Eckhoff et al., 2025; Zhou et al., 2024), while graph-based approaches have demonstrated the value of relational and temporal context for multi-stage attack analysis (Cheng et al., 2024; Hassan et al., 2020). Research on deployment, calibration, and explainability also shows why predictive performance alone is insufficient when learned models influence operational decisions (Apruzzese et al., 2023; Guo et al., 2017; Ribeiro et al., 2016).

What remains less clearly defined is how these capabilities should be integrated across an existing production detection-and-response lifecycle while retaining the control required of a production security system. NS-D&R addresses this as a question of authority: what should neural and symbolic components be permitted to do within an already functioning security operations pipeline?

The architecture makes four contributions. First, it introduces bounded neural authority. Neural models can identify patterns, retrieve similar activity, rank evidence, and propose hypotheses, but model confidence does not itself confer authority over production decisions.

Second, existing deterministic mechanisms remain authoritative. Human-authored detections, rules, schema constraints, policy controls, and the alerting stage continue to govern production outcomes. Neural systems can contribute evidence and proposals, but cannot bypass these controls.

Third, this division of responsibility applies across the detection-and-response lifecycle, including enrichment, scoring, persistence, clustering, alerting, triage, and feedback, while the existing detection pipeline continues to operate.

Finally, NS-D&R provides deterministic fallback. Neural components augment rather than replace the existing pipeline; if a model becomes unavailable or unreliable, its influence can be removed while core detection continues from the deterministic baseline.

NS-D&R therefore contributes an architectural model rather than a new neural model, symbolic formalism, or anomaly-detection algorithm. It defines how neural and symbolic capabilities can be incorporated into production security operations while preserving explicit authority boundaries, governance, and failure behaviour.

3. System Model: The Signal-Based Detection Pipeline

We assume a baseline detection pipeline that reflects the general structure of a contemporary SOC. Although implementations differ between organisations and products, the underlying process is broadly similar: security telemetry is collected and normalised, detection logic identifies suspicious activity, additional context is applied, related findings are brought together, and the resulting alerts are presented to analysts for investigation. Event correlation is an established part of this process, particularly where activity must be linked across multiple sources or reconstructed as part of a multi-stage attack (Kotenko et al., 2022).

The baseline pipeline is divided into the following stages:

1. **Telemetry ingestion.** Raw security telemetry is collected from heterogeneous sources, including endpoints, identity systems, networks, cloud platforms, and applications.
2. **Schema normalization.** Events are mapped to a common security-event schema so that downstream components can work with a consistent set of fields and entities.
3. **Detection analytics.** Human-authored queries and rules evaluate normalized events and produce structured *detection findings*. The underlying compute platform is interchangeable and may include streaming, batch-processing, or log-search systems.
4. **Enrichment.** Findings are supplemented with information about assets, identities, vulnerabilities, threat intelligence, and other operational context.
5. **Risk rules.** Deterministic rules adjust the risk attached to a finding based on factors such as asset criticality, known-benign activity, policy, or other contextual information.
6. **Persistence.** Findings and relevant intermediate artifacts are retained in an analytics datastore so they remain available for correlation, investigation, and historical analysis.
7. **Clustering.** Findings that share evidence, entities, or other relevant characteristics are grouped into clusters intended to approximate incidents.
8. **Alerting.** The final stage applies thresholding, deduplication, and other alerting criteria before sending selected activity to a case-management or ticketing system.
9. **Triage and feedback.** Analysts investigate alerts, follow response playbooks, record outcomes, and feed what they learn back into the detection process through tuning requests, score adjustments, and new or modified rules.

This abstraction is deliberately vendor-neutral. The purpose is not to prescribe a particular SOC technology stack, but to establish the deterministic pipeline that the neuro-symbolic components introduced later in the paper will augment.

3.1 Anatomy of the Detection Analytics Component

Detection analytics combines different methods to turn security events into findings. Signature and pattern matching are effective for known activity, while anomaly and behavioural approaches can identify previously unseen behaviour but generally introduce greater uncertainty and false positives (García-Teodoro et al., 2009).

For NS-D&R, detection analytics is divided into three broad sub-components. The first two perform routine detection, while the third connects activity across sources and provides higher-level context.

3.1.1 String- and pattern-based search

The first sub-component evaluates events against predefined patterns, including matches against normalized fields, regular expressions, file hashes, indicators, and signature languages such as YARA. This approach is effective when malicious artifacts or behaviours can be described precisely, but depends on appropriate signatures being available (García-Teodoro et al., 2009). The same design creates its main limitation: activity that does not match a defined pattern may be missed, and small changes can evade narrowly written signatures. Signature content therefore requires continued maintenance as attacks and indicators change (Hubballi & Suryanarayanan, 2014). Within NS-D&R, this sub-component provides atomic findings and literal artifacts that later embedding and similarity components can generalize beyond.

3.1.2 Temporal, statistical, and threshold analytics

The second sub-component covers analytics that reason across multiple events, over time, or relative to an expected pattern or defined limit. These methods require state or historical context rather than evaluating each event independently.

Temporal and sequence analytics connect activity within a defined period, either in a particular order or within the same window. This is important for multi-stage attacks, where individual events may not reveal the wider activity (Kotenko et al., 2022).

Statistical baselining models expected behaviour and identifies significant deviations. Statistical and machine-learning methods can therefore expose activity without a known signature, although anomaly-based detection generally introduces greater uncertainty and false-positive rates than signature matching (García-Teodoro et al., 2009).

Threshold, aggregation, and rarity analytics apply explicit limits to activity observed over time, such as event counts, rates, distinct values, rare combinations, or defined sequences. Unlike learned baselines, these boundaries are set by the detection engineer, making the logic straightforward to inspect and reproduce but less able to capture variations in normal behaviour.

These techniques are grouped together because they operate over the same normalized event stream and rely on similar windowing and state-management capabilities. A detection may also combine them: activity can satisfy a threshold, complete a sequence, and deviate from an established baseline. The neural entity representations introduced in Section 6.2 extend this model by learning relationships that fixed thresholds and manually defined baselines may not capture.

3.1.3 Multi-source correlation analytics

The third sub-component connects events and findings across telemetry sources. An identity event, for example, may become more significant when associated with an endpoint process and subsequent cloud action involving the same user or resource. Security-event correlation establishes these relationships and can help identify the progression of multi-step attacks (Kotenko et al., 2022).

Cross-source correlation also depends on normalization and enrichment. Security systems may represent the same entity differently, requiring common identifiers or entity resolution before events can be reliably joined.

In the baseline pipeline, these relationships are expressed through predefined correlation logic. In NS-D&R, this provides the deterministic precursor to the detection knowledge graph described in Section 6, which turns individual correlation joins into persistent relationships that can be queried and used by learned models.

Execution can occur over different time horizons. Atomic detections and short-window correlations are suited to real-time or near-real-time analysis, while behavioural and complex cross-source analytics may require longer historical context (Kotenko et al., 2022). The pipeline can therefore use different execution tiers rather than requiring every analytic method to operate under the same latency constraints.

Figure 1 summarises this decomposition.

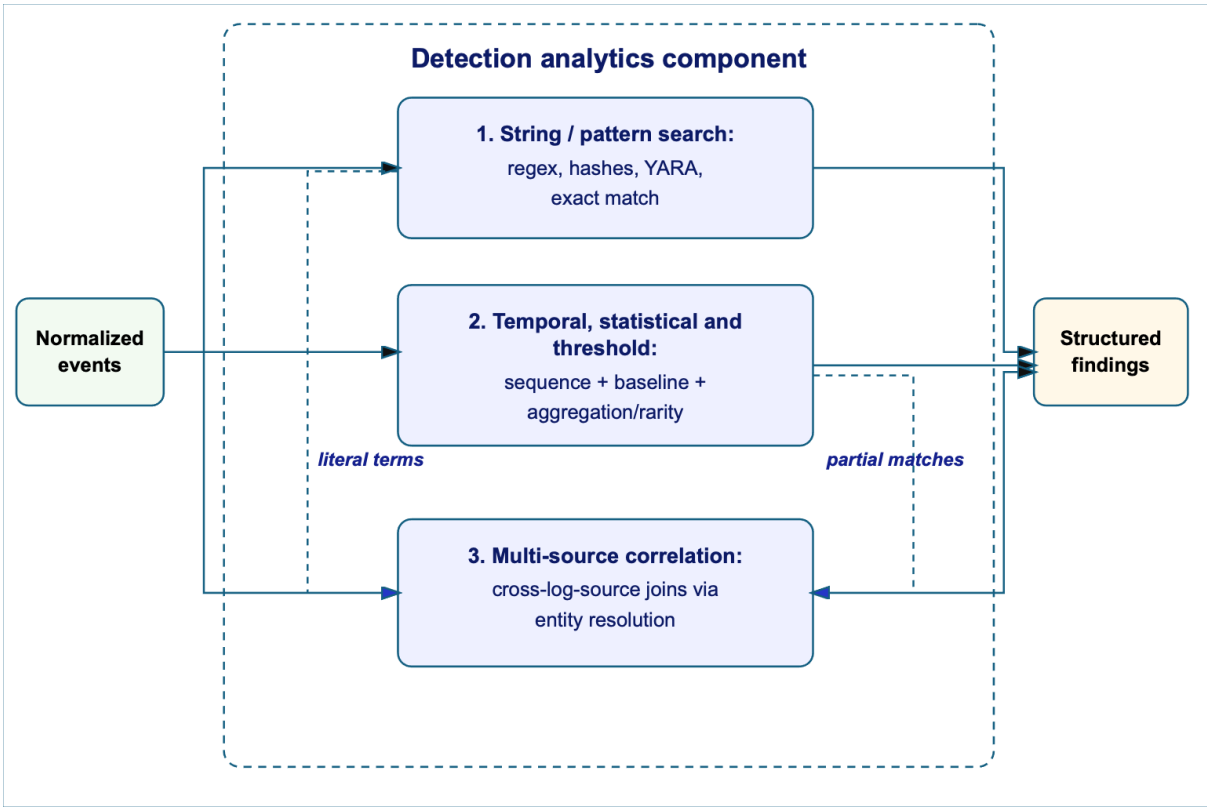

Figure 1. Anatomy of the detection analytics component. Three sub-components consume the same normalized event stream and emit structured findings: pattern matching for known artifacts, a combined temporal/statistical/threshold family for behavior over time, and multi-source correlation for high-fidelity cross-source joins. Correlation consumes the outputs of the first two and is the deterministic precursor to the detection knowledge graph.

Table 1. Detection Analytics Sub-components and Their Operational Characteristics

Sub-component	Primary technique	Detects	Nature and trade-off
String / pattern search	Exact match, regex, hashes, YARA	Known artifacts and signatures	Deterministic, low false-positive, real-time; reactive and easily evaded by variation
Temporal, statistical and threshold	Sequence/temporal correlation, learned baselines, aggregation and rarity	Multi-stage progressions, deviations from norm,	Stateful and windowed; captures temporal and behavioural variation; learned baselines can

		volume/rate/rarity outliers	add noise, while fixed thresholds may miss variation
Multi-source correlation	Cross-source joins on resolved entities	Attack patterns spanning multiple log sources	High-fidelity, fewer alerts; depends on entity resolution and schema quality

Two properties of this baseline are essential and must be preserved by any augmentation: (i) detection logic is human-authored, versioned, and reviewable; and (ii) every alert is traceable to the findings and rules that produced it. The architecture in this paper is designed to add neural capability without sacrificing either property.

4. Architectural Principles

NS-D&R is built around a simple division of responsibility:

Neural models discover, embed, rank, summarize, and hypothesize; symbolic components validate, constrain, explain, deduplicate, score, and govern.

Neural models suit patterns, generalization, retrieval, and hypothesis generation; symbolic components suit decisions that must remain explicit, reproducible, and subject to policy. This separation reflects broader guidance on trustworthy AI, where accountability, transparency, and defined human-AI responsibilities are treated as properties of the system as a whole rather than of the model alone (Tabassi, 2023).

This can be expressed formally. Let x be the evidence available for a finding, N the neural components, C_n the context they produce, S the symbolic reasoning layer, and D the resulting production decision:

$$C_n = N(x) \qquad (1)$$

$$D = S(x, C_n) \qquad (2)$$

Neural output therefore influences D only through S, which may accept, constrain, modify, or discard its contribution according to applicable rules and policy. Neural evidence therefore has no independent production authority.

Three design commitments follow. First, NS-D&R augments the existing detection pipeline rather than replacing it: detection findings remain the atomic unit, deterministic risk rules remain the production guardrail, and alerting remains where findings become operational alerts. Neural outputs — embeddings, anomaly scores, similarity results, summaries, hypotheses — are persisted as evidence alongside findings for symbolic components to use during scoring, clustering, investigation, and response; they are never emitted as alerts independently. Second, neural effect on scoring is explicitly bounded: a model can strengthen or weaken a finding's evidence only within symbolic limits, since neural confidence is not necessarily well calibrated to correctness (Guo et al., 2017), and risk-management guidance likewise emphasises defined risk tolerances and clear human-AI responsibility (Tabassi, 2023). Third, decisions must be reconstructable after the fact: the system retains the facts used, the model and rule versions involved, the reasoning performed, and any resulting risk change, so that both the decision and the logic behind it can be recovered. This matters wherever AI operates inside consequential workflows (Tabassi, 2023).

Together, these commitments set the boundary the rest of the architecture follows: neural components broaden what the system can see and infer, but production authority remains constrained by explicit rules, policy, and human oversight.

5. The NS-D&R Architecture

NS-D&R is organised into three planes: detection, reasoning, and response, each defining a different role for neural and symbolic components.

The detection plane retains the existing detection pipeline as the production authority. Neural components described in Section 6 augment the deterministic analytics from Section 3.1 with entity anomaly scores, behavioural embeddings, historical similarity, attack-stage hypotheses, and candidate relationships between findings. This combines the generalisation capabilities of learned representations with explicit knowledge and reasoning (Hitzler et al., 2022).

The reasoning plane determines how this context can influence production decisions. It operates over the normalised schema, evidence, detection identifiers, adversary-technique mappings, known-benign patterns, risk rules, asset and identity context, and historical outcomes. Graph-based research demonstrates the value of preserving relationships between security events and entities for multi-stage attack reasoning (Cheng et al., 2024; Hassan et al., 2020). NS-D&R extends this principle by using the reasoning plane to constrain neural influence over scoring and clustering.

The response plane supports analysts through summarisation, triage explanation, timeline construction, investigative pivots, playbook selection, and detection-tuning suggestions. These functions may be invoked during investigation rather than forming a mandatory sequential pipeline stage. Actions with operational consequences remain subject to human approval (Tabassi, 2023).

Operationally, normalised telemetry passes through existing detection analytics to produce structured findings. A neuro-symbolic context layer then applies entity graphs, embeddings, behavioural and anomaly models, symbolic rules, hypothesis generation, and causal or attack-path reasoning. Risk adjustment determines how much of this context can affect production risk: symbolic adjustments remain deterministic, while neural contributions are bounded. Findings and their supporting context and reasoning are then persisted for analysis and audit.

Clustering uses both deterministic relationships and neuro-symbolic context to identify connections between findings while retaining the deterministic path as a baseline. The existing alerting stage remains responsible for determining which clusters become production alerts. Once alerts reach analysts, response-plane capabilities assist investigation and triage without independently executing consequential actions.

Investigation outcomes, tuning requests, and detection changes feed back into later decisions. Historical outcomes can inform neural representations and similarity models, while confirmed patterns and analyst decisions can inform symbolic rules and policies. This allows both neural and symbolic components to evolve without changing the underlying division of authority.

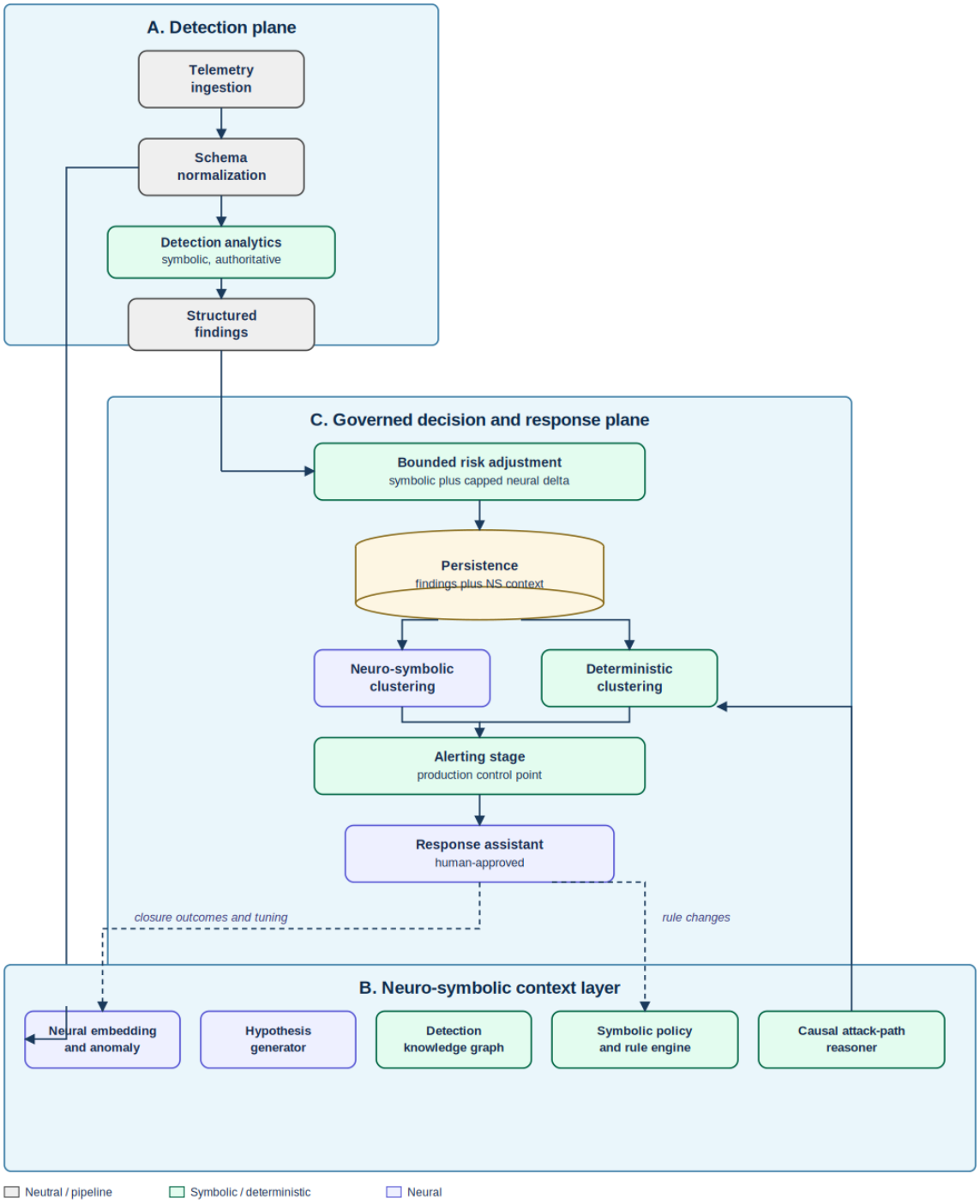

Figure 2. End-to-end NS-D&R architecture. Neural components (blue) discover, embed, and hypothesize; symbolic components (green) validate, score, and govern. The deterministic detection and alerting stages remain the production control points.

6. NS-D&R Architectural Components

The NS-D&R architecture brings together a set of neural and symbolic components, each responsible for a different part of the detection-and-response process. These components are not intended to operate as separate detection pipelines. Instead, they build on the architecture described in Section 5 by adding relational context, learned representations, controlled adjustments to risk, reasoning across related findings, and support for analyst investigation. The following sections describe how each component works, what information it uses and produces, and the limits placed on its influence over production decisions.

6.1 Detection Knowledge Graph

The first symbolic component is a detection knowledge graph built from artifacts already present in the detection-and-response pipeline. Graph-based approaches are useful in security because relationships between events and entities can reveal activity that is difficult to identify from individual events. Provenance-based systems have demonstrated this value in reconstructing multi-stage attacks across entities and time (Han et al., 2020; Hassan et al., 2020; Cheng et al., 2024).

In NS-D&R, the graph extends beyond system provenance. Nodes represent detections, findings, alert clusters, evidence entities, and operational concepts such as risk rules, adversary techniques, playbooks, tuning actions, and investigation outcomes. Edges record how these objects relate: detections produce findings, findings contain evidence and map to entities or techniques, rules adjust findings, findings form clusters, and clusters generate alerts. The graph therefore captures both security activity and how the detection system interpreted and acted on it.

This provides a persistent extension of the multi-source correlation described in Section 3.1.3. Relationships that would otherwise exist only as joins during detection become durable objects that can be queried and reused. Unlike conventional provenance graphs, NS-D&R also incorporates detection and operational artifacts alongside the underlying security evidence.

The graph also provides an interface between learned representations and explicit reasoning. Neural representations can operate over entities and relationships defined symbolically, while subsequent reasoning remains tied to those explicit relationships. This reflects the broader neuro-symbolic approach of combining learned representations with interpretable symbolic structure (Barbiero et al., 2023).

In practice, the graph supports both symbolic and learned analysis. Explicit relationships can answer questions such as whether an activity chain has been observed before, whether activity matches a known-benign workflow, or which rule changed a finding's risk. The same structure can support graph representation learning over nodes and relationships (Hamilton et al., 2017), including the identification of anomalous or recurring structures and candidate attack paths (Cheng et al., 2024). Learned relationships remain proposals: symbolic constraints and existing evidence determine whether they can influence production reasoning.

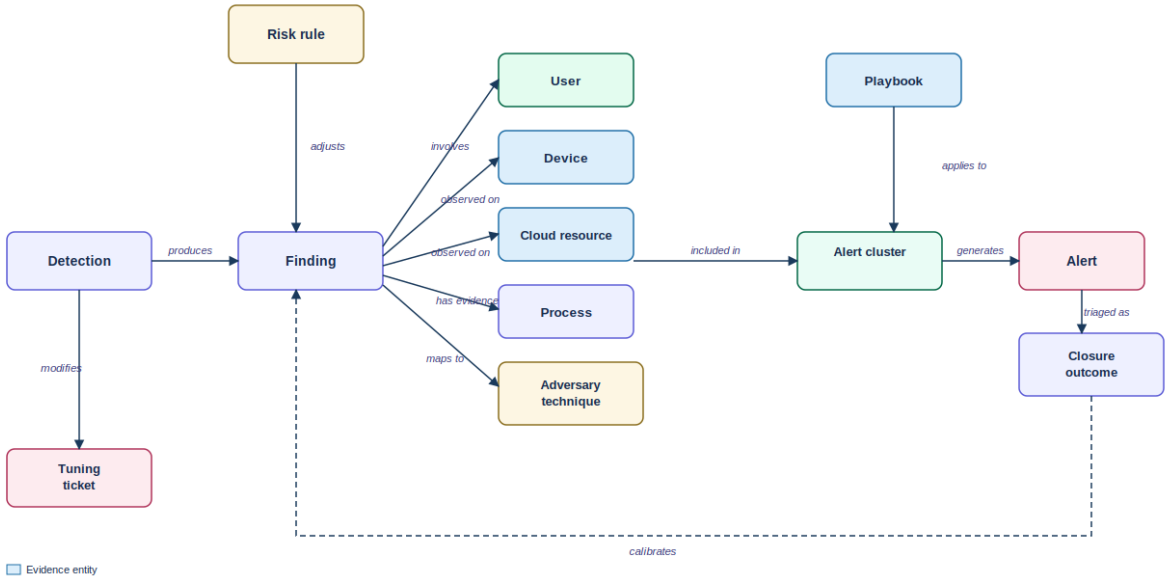

Figure 3. Detection knowledge graph. The graph connects detections and findings with evidence entities, alert clusters, risk rules, playbooks, tuning activity, and closure outcomes. Typed relationships preserve how findings are produced, enriched, clustered, alerted, and subsequently used to calibrate detection logic.

6.2 Neural Entity Representation and Anomaly Scoring

For each major evidence entity—such as users, devices, process command lines, cloud resources, repositories, network endpoints, alert clusters, and detection sequences—the

system maintains representations over relevant time windows. These representations provide additional context about the activity associated with an entity and how that activity compares with its own history, its peers, and previous security incidents. Representative outputs include an entity-level anomaly score, a measure of peer-group deviation, and a set of similar historical alert clusters together with their similarity scores and eventual closure outcomes. Learned representations have been used to capture relationships between entities and support anomaly detection in complex security data, including graph-based detection systems (Hamilton et al., 2017; Cheng et al., 2024).

This allows the detection pipeline to ask questions that are difficult to express through deterministic rules alone: Which previous incidents most closely resemble this one, and how were they resolved? or How unusual is this activity for the entity itself and compared with similar entities? The resulting context can help distinguish genuinely unusual behaviour from activity that is uncommon globally but normal for a particular user, device, or workload.

NS-D&R does not prescribe a particular model for producing these representations. Sequence models, autoencoders, isolation forests, or contrastive encoders could be used for behavioural anomaly detection; text or code embeddings could represent the semantics of command lines and other textual evidence; and embedding-based retrieval could be used to find similar historical incidents. Isolation forests provide an established approach to unsupervised anomaly detection (Liu et al., 2008), while autoencoder-based methods provide another means of identifying anomalous observations in high-dimensional data (Zhou & Paffenroth, 2017). The choice of model can therefore change as techniques improve without changing the surrounding architecture. What NS-D&R defines is how the resulting evidence is represented, retained, and allowed to influence production decisions.

These learned representations complement the deterministic temporal, statistical, and threshold analytics described in Section 3.1.2. Rather than relying only on fixed baselines and operator-defined thresholds, they can learn behavioural structure from historical activity and

identify relationships that have not been explicitly encoded in advance. They also build naturally on the pattern-based analytics described in Section 3.1.1: literal terms and other features identified during pattern matching can become inputs to command-line and text representations. The neural layer therefore extends the context available to the existing detection pipeline rather than creating a separate source of production detections.

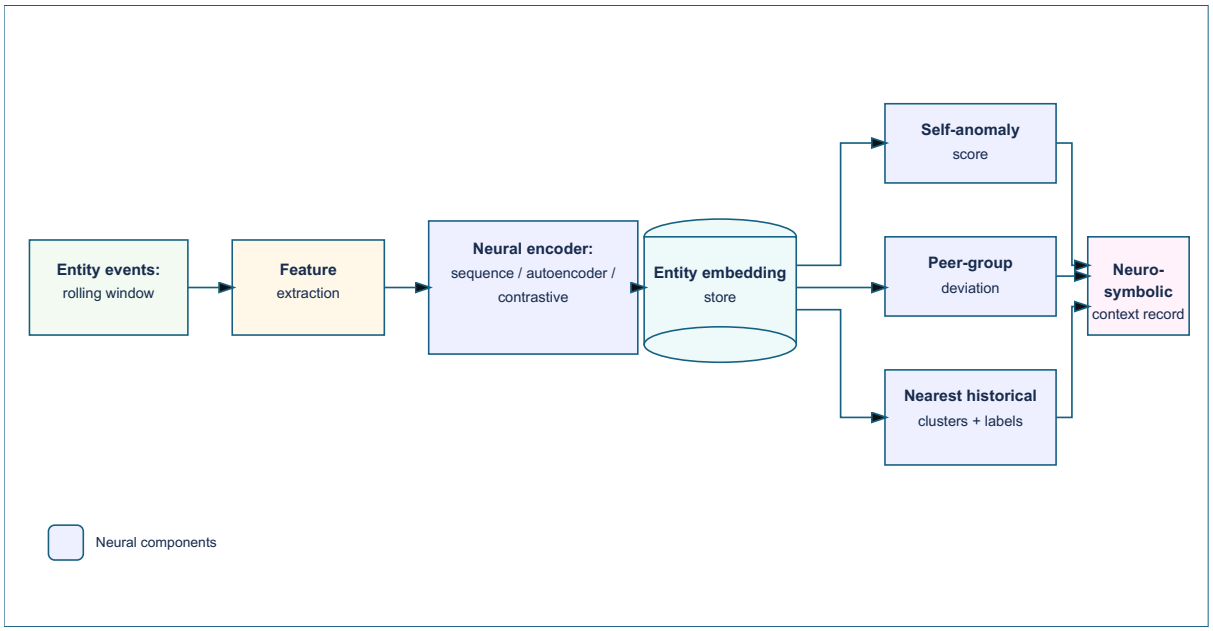

Figure 4. Neural entity representation. Rolling-window entity events are encoded into learned representations that produce self-anomaly, peer-group deviation, and historical-similarity signals. These outputs are stored in the neuro-symbolic context record for use by downstream reasoning and scoring components.

6.3 Neuro-Symbolic Enrichment

Enrichment is extended so that each finding carries a structured neuro-symbolic context record before clustering. The record can include an entity anomaly score, similarity to historical true- and false-positive activity, peer-group deviation, candidate attack-stage hypotheses, and links to relevant historical incidents. It also retains a small set of human-readable facts explaining the additional context—for example, that a user executed a rare process on a new device or that a related suspicious authentication occurred shortly beforehand. Providing interpretable evidence alongside model outputs can make those outputs more useful to human decision-makers (Ribeiro et al., 2016).

A strict boundary remains between enrichment and decision-making. Neural context can add evidence to a finding, but it cannot directly suppress an alert or increase its production severity. Instead, it is consumed by the symbolic scoring, clustering, and response components, which determine whether and how it can influence the outcome. This separation is particularly important because model confidence does not necessarily correspond to correctness (Guo et al., 2017).

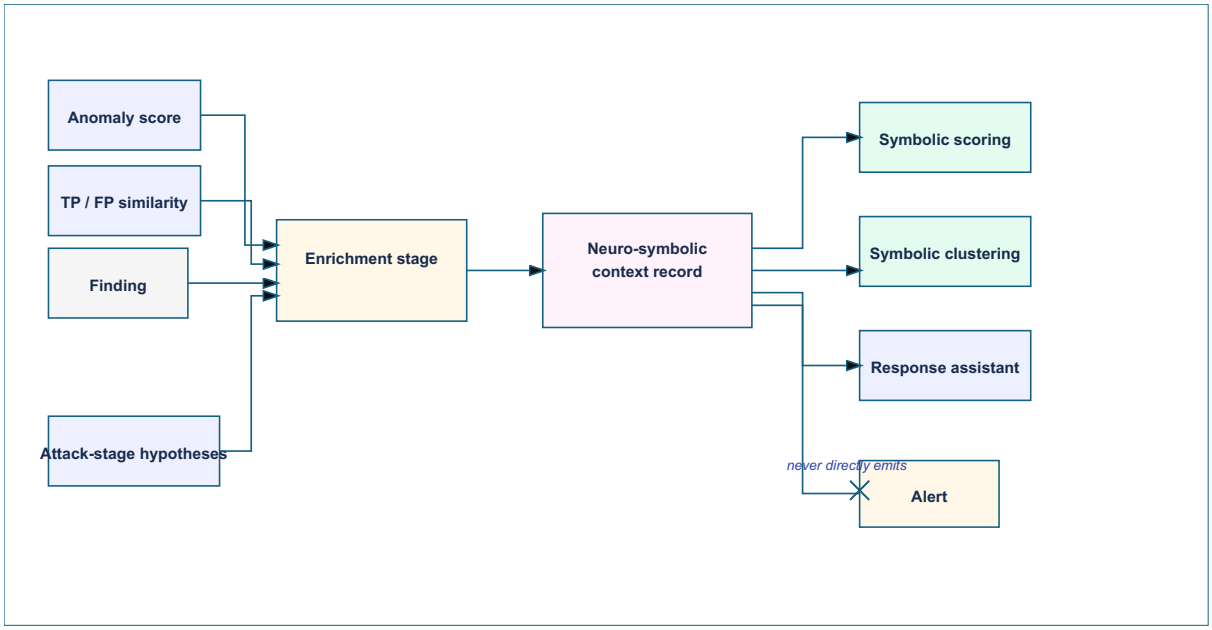

Figure 5. Neuro-symbolic enrichment. Neural signals and finding data are combined during enrichment and stored in a neuro-symbolic context record. The record is available to symbolic scoring, symbolic clustering, and the response assistant, but cannot directly generate an alert.

6.4 Symbolic Risk Reasoner

The existing risk-rule engine already provides a symbolic layer, so NS-D&R builds on it rather than creating a separate scoring mechanism. Neural context can be used as an input to these rules, but the resulting adjustment remains a fixed, policy-defined value rather than being scaled by the neural score. For example, risk might increase only when anomaly and true-positive similarity scores cross defined thresholds and the detection belongs to an approved set.

The same approach can raise risk when several signals point towards malicious activity, reduce it when behaviour resembles known benign patterns, or account for unusual peer-group behaviour and previous analyst outcomes. The neural layer provides additional context; the symbolic rules decide what to do with it. This keeps the final scoring decision predictable, explainable, and auditable.

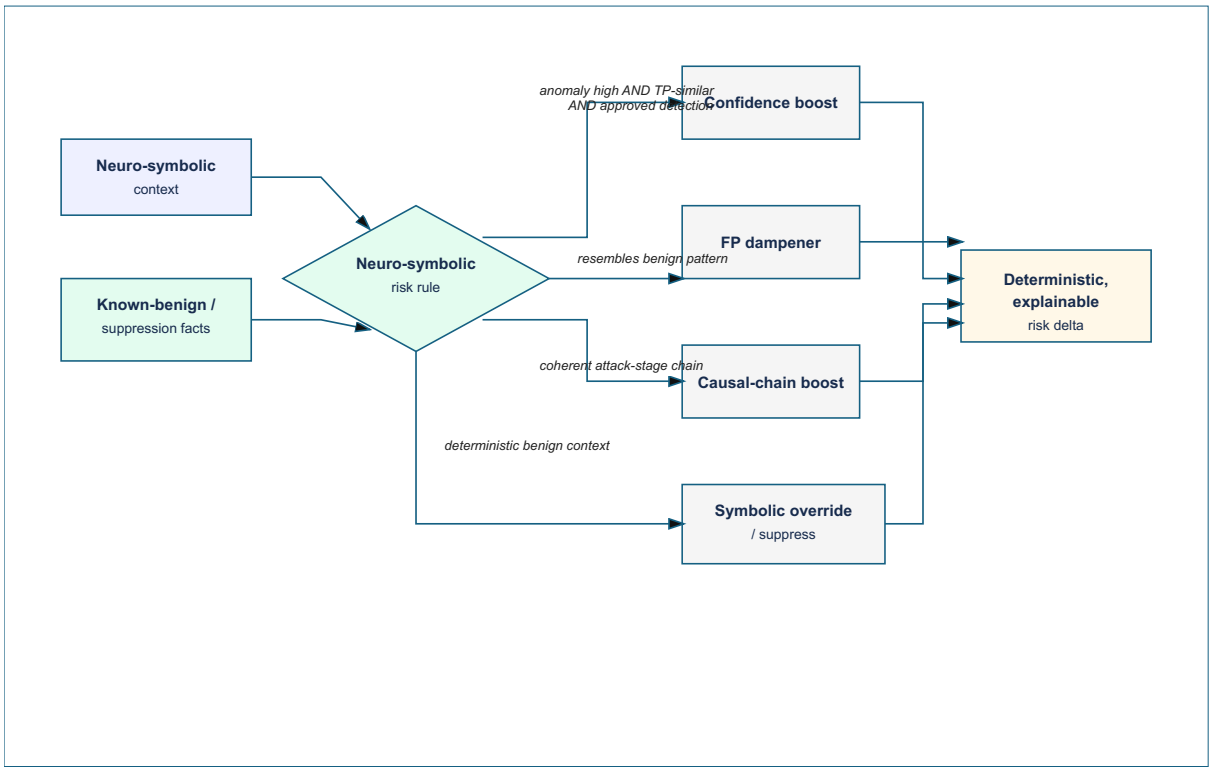

Figure 6. Symbolic risk reasoner. Neuro-symbolic rules consume neural context and benign facts but always emit a deterministic, explainable risk delta, keeping the decision auditable.

6.5 Bounded Confidence Calibration

Risk scoring combines the existing human-defined base score with symbolic rule adjustments and additional context derived from neural evidence, historical outcomes, causal relationships, and known-benign activity. The key constraint is that the neural contribution is bounded: it can influence borderline cases, but it cannot dominate the final production score.

Where neural evidence satisfies a symbolic rule condition, the resulting symbolic adjustment is a fixed policy-defined value rather than a value scaled by the neural score. The same

applies to adjustments based on historical outcomes or validated causal relationships; neural evidence may determine whether an applicable symbolic condition is satisfied, but not the magnitude of the resulting adjustment.

Let R_b denote the human-authored base risk, Δ_s the symbolic rule adjustment, Δ_h the historical-fidelity adjustment, Δ_c the validated causal-chain adjustment, $\Delta_b \geq 0$ the magnitude of a deterministic known-benign reduction, and $\Delta_{n,}$ the neural-context adjustment. The maximum permitted neural contribution is defined by λ. The bounded neural adjustment is therefore:

$$\hat{\Delta}_n = \text{clip}(\Delta_{n,} - \lambda, \lambda) \qquad (3)$$

The final risk score Rf is then calculated within the permitted scoring range $[R_{min,}R_{max}]$:

$$R_f = \text{clip}(R_b + \Delta_s + \Delta_h + \Delta_c - \Delta_b + \hat{\Delta}_{n,}R_{min,}R_{max}) \qquad (4)$$

This allows neural context to affect the relative risk of a finding while placing an explicit limit on how much it can change the production score. The value of λ is a policy parameter and can be adjusted according to the organisation's tolerance for neural influence. Two additional safety invariants constrain how neural evidence can affect alerting:

$$Suppress(A, N) = 0 \qquad (5)$$

$$\text{Severity}(A, N) \neq S_{max} \qquad (6)$$

Here, A denotes an otherwise qualifying finding or alert, N denotes neural evidence considered without independent symbolic justification, and S_{max} denotes the highest permitted production severity. Equations (5) and (6) specify architectural constraints rather than probabilistic relationships: neural evidence without independent symbolic justification cannot suppress an otherwise qualifying alert or independently assign the highest production severity.

The architecture also provides a deterministic fallback when a neural component becomes unavailable, drifts beyond an acceptable threshold, or its output is otherwise considered unreliable. In these circumstances, the bounded neural contribution $\hat{\Delta}_{n,}$ is set to zero, leaving the final score to be determined by R_b, Δ_s, Δ_h, Δ_c, and Δ_b. The production scoring path therefore remains available without neural input.

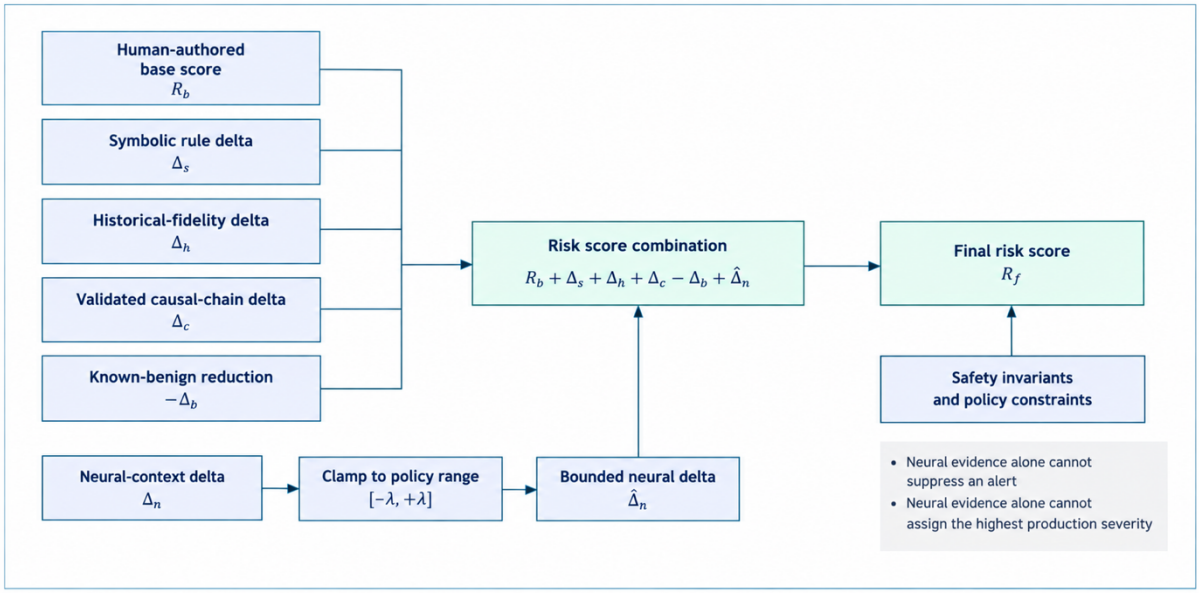

Figure 7. Bounded confidence calibration. The neural-context delta is clamped to a policy-defined range before being combined with the base score and other risk adjustments. Safety invariants then prevent neural evidence alone from suppressing an alert or assigning the highest production severity.

6.6 Neuro-Symbolic Clustering

Deterministic clustering remains the baseline, grouping findings through shared identifiers such as users, devices, processes, or cloud resources. NS-D&R adds three neural-assisted modes alongside it.

Semantic cluster expansion uses embeddings to identify related findings and historically similar incidents, but treats these as candidate links rather than automatically merging them. *Causal-chain clustering* allows neural models to propose relationships between findings, which are then checked against symbolic constraints such as timing, shared or linked entities, compatible adversary tactics or techniques, and evidence continuity. *Multi-entity incident clustering* extends grouping beyond a single entity, allowing activity across users, devices, cloud accounts, repositories, network addresses, and applications to be considered part of the same incident.

These methods complement existing graph- and provenance-based approaches that use relational and temporal context to connect activity across multi-stage attacks (Cheng et al., 2024; Hassan et al., 2020).

The neural-proposal and symbolic-validation pattern can be illustrated formally for causal-chain clustering. For two findings $f_i$and f_j, the neural clustering component may propose a candidate relationship:

$$e_{ij}^{N} = (f_i, f_j, c_{ij}) \qquad (7)$$

where $c_{ij} \in [0,1]$ represents the neural relationship score assigned to the proposed edge, with higher values indicating stronger learned evidence that the two findings are related. The score expresses the strength of the neural proposal but does not determine whether the edge is accepted; it may be retained for ranking, analyst review, and model evaluation. The relationship remains a candidate until it passes symbolic validation:

$$V\left(e_{ij}^{N}\right) = T_{ij} \wedge E_{ij} \wedge A_{ij} \wedge L_{ij} \qquad (8)$$

Here, T_{ij} represents temporal compatibility, E_{ij}a shared or linked entity relationship, A_{ij}compatibility between the associated adversary tactics or techniques, and L_{ij}evidence continuity between the findings, which may be established through shared session, device, process, network, or other required contextual evidence. Each validation predicate is Boolean, such that $T_{ij}, E_{ij}, A_{ij}, L_{ij} \in \{0,1\}$, where 1 indicates that the corresponding symbolic condition is satisfied and 0 that it is not.

In this illustrative validator, all four conditions are required for acceptance. Other relationship types may use different sets of symbolic predicates, but the same principle applies: the conditions required by the applicable validation policy must be satisfied before a neural proposal becomes an operational edge.

The proposed relationship becomes an accepted operational edge only when these constraints are satisfied:

$$e_{ij}^{*} = \begin{cases} e_{ij}^{N} & \text{if } \mathrm{V}\left(e_{ij}^{N}\right) = 1 \\ \emptyset & \text{if } \mathrm{V}\left(e_{ij}^{N}\right) = 0 \end{cases} \qquad (9)$$

This formalises the same authority boundary used throughout NS-D&R: the neural component may propose a relationship, but it does not establish that relationship as an operational fact. The symbolic layer determines whether the proposed edge can influence clustering. Importantly, neuro-symbolic clustering remains upstream of the existing alerting stage. It supplies additional clustered findings for consideration but does not replace the production alerting mechanism.

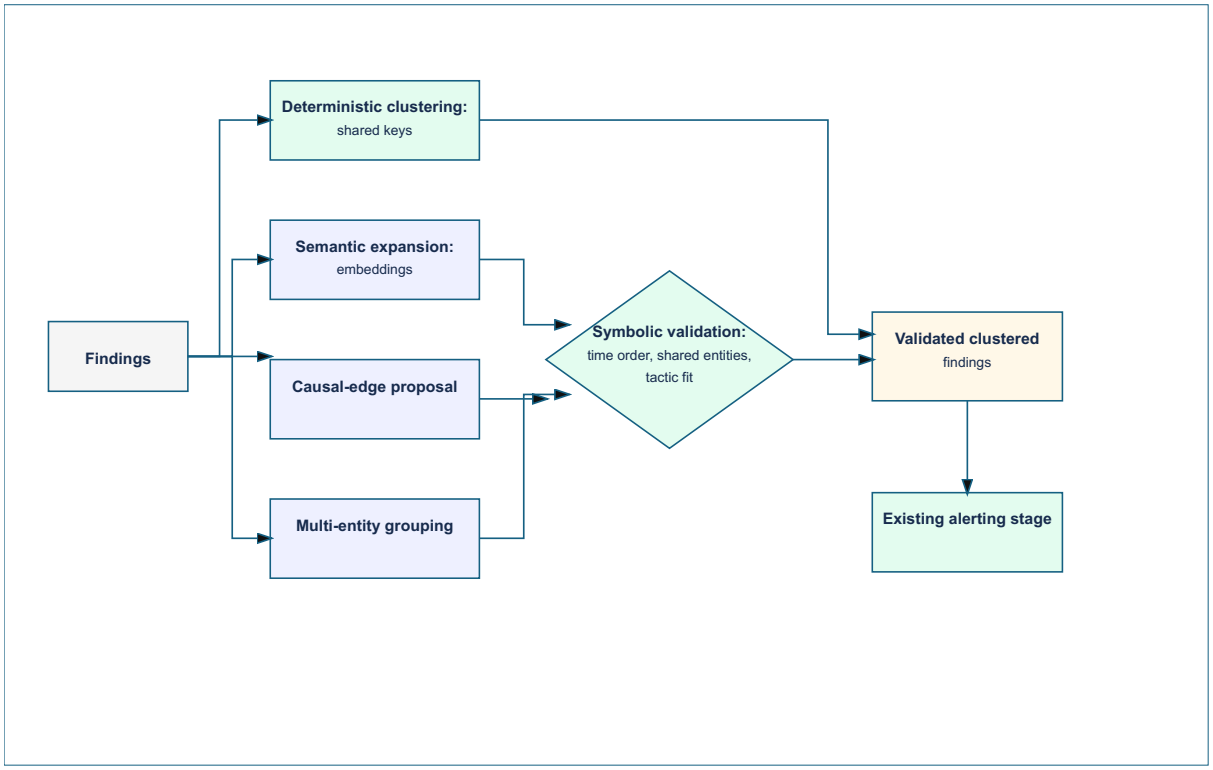

Figure 8. Neuro-symbolic clustering. Neural components propose semantic, causal, and multi-entity relationships between findings, which are subject to symbolic validation before contributing to validated clusters. Deterministic clustering remains available in parallel, and the resulting clustered findings feed the existing alerting stage.

Appendix A illustrates how these components can operate together across a vulnerability-to-detection lifecycle, from advisory ingestion and environmental relevance assessment through detection generation, validation, and human review.

7. Persistence Model

Neural context needs to be stored in a way that makes it auditable without adding unnecessary complexity to the existing finding schema. NS-D&R therefore uses a small set of adjacent tables. These store the neuro-symbolic context associated with each finding, entity embeddings and feature summaries, proposed and validated graph edges, validated clusters and their explanations, and the information used to explain an alert. The alert-level record can include the primary hypothesis, supporting and contradicting evidence, recommended playbooks or investigative pivots, similar historical alerts, and any subsequent analyst feedback. Model and data versions are retained throughout, allowing the context behind a decision to be reconstructed and reviewed later.

8. The Response Plane: A Neuro-Symbolic Triage Assistant

On the response side, NS-D&R begins with an analyst-facing assistant rather than an autonomous responder. The assistant draws on the alert cluster, its findings and evidence, neuro-symbolic context, graph relationships, matched risk rules, historical outcomes, relevant playbooks, and asset and identity information. Its purpose is to help answer the questions analysts typically face during triage: what happened, why the alert fired, which findings matter most, what hypothesis best fits the evidence, which playbook applies, what to investigate next, and whether the activity resembles previous true or false positives. The assistant's outputs remain advisory context rather than production decisions, with consequential decisions retained by the human analyst.

The assistant draws its hypotheses from the symbolic evidence available in the normalised schema and knowledge graph, keeping them tied to the evidence available to the system. A response might identify a potentially compromised developer account carrying out discovery and repository access. Supporting evidence could include several findings for the same user within a short period, an unusual repository clone, a cloud role assumption, a high peer-relative anomaly score, and similarity to previous true-positive incidents. At the same time, it might identify evidence against that hypothesis, such as a previously observed source address and a healthy managed device.

From this evidence, the assistant can recommend an appropriate investigation playbook and useful pivots, such as reviewing authorization grants, repository activity, cloud role assumptions, location context, and relevant change records. Presenting both supporting and contradicting evidence gives the analyst a clear basis for assessing, challenging, or rejecting the recommendation.

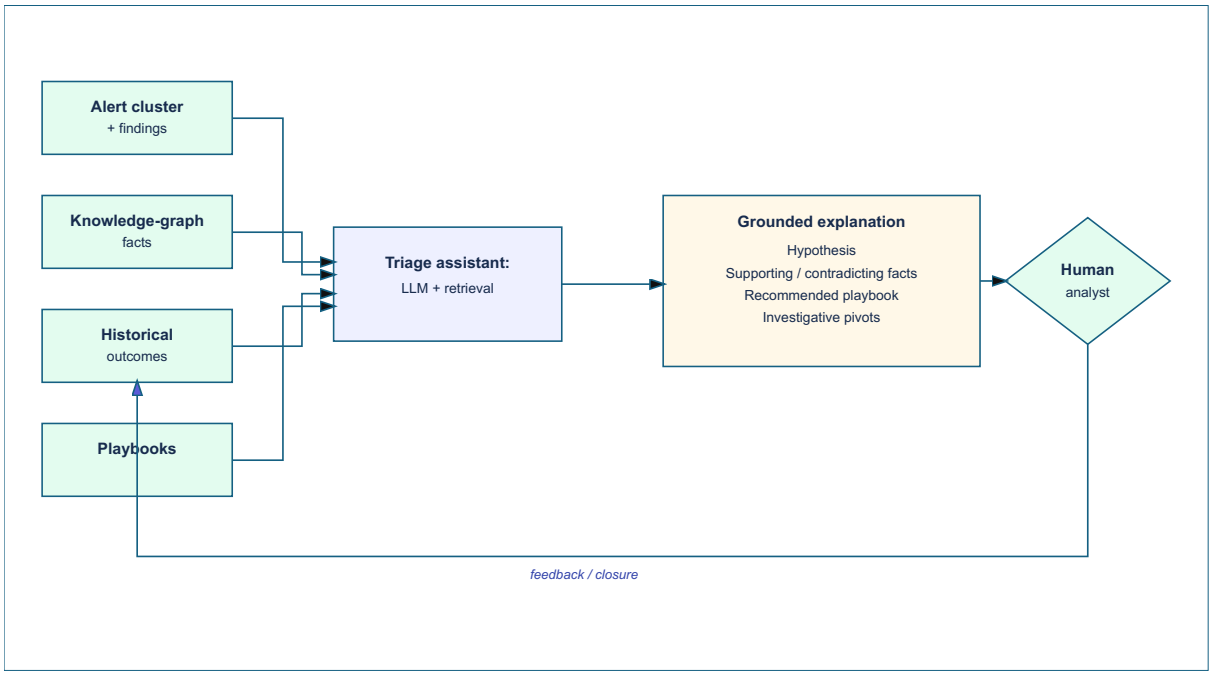

Figure 9. Neuro-symbolic triage assistant. The assistant combines alert findings with knowledge-graph facts, historical outcomes, and applicable playbooks to produce a grounded explanation containing the primary hypothesis, supporting and contradicting evidence, recommended playbook, and investigative pivots. The human analyst remains the decision point, with closure outcomes feeding back into the system.

9. Division of Labour

The architecture gives neural and symbolic methods different responsibilities. Neural components are used where the system benefits from pattern recognition, similarity, anomaly detection, prediction, and summarisation. This includes retrieving similar incidents, identifying unusual entity behaviour, comparing command-line semantics, generating triage hypotheses, predicting attack relationships, prioritising alerts, and suggesting detection tuning.

Symbolic components remain responsible for decisions that need to be explicit and reproducible. These include schema validation, risk-rule execution, risk scoring, allow and block lists, suppression handling, evidence matching, adversary-technique constraints, deduplication, approval gates, response authorisation, and auditability.

The separation is deliberate: neural methods broaden what the system can recognise and infer, while symbolic controls determine how that information is allowed to affect production decisions.

10. Adoption Roadmap

NS-D&R is intended to be introduced gradually, allowing each stage to provide useful capability without giving neural components too much authority too early. Phase 0 establishes the data contracts: the neuro-symbolic context schema, permitted model-generated fields, limits on neural score contributions, versioning, and audit requirements.

Phase 1 adds passive enrichment, including embedding storage, historical alert similarity, and basic explanations, without changing risk scores or alerting.

Phase 2 introduces human-reviewed recommendations for risk changes and detection tuning.

Phase 3 allows bounded neural contributions to scoring, while retaining the safeguards that neural evidence alone cannot suppress an alert or raise it to the highest severity.

Phase 4 introduces neuro-symbolic clustering through semantic similarity, causal relationships, and multi-entity incidents.

Phase 5 adds the response assistant, including timeline summaries, explanations, playbook and investigation recommendations, historical comparisons, and analyst feedback.

A practical first implementation does not require the entire architecture. Three capabilities provide a relatively low-risk starting point: retrieving similar historical true- and false-

positive incidents, adding entity anomaly scores for users, devices, and cloud resources, and using neural models to propose causal relationships that are validated symbolically before affecting clustering.

11. Governance and Auditability

Governance is built into the architecture rather than added after deployment. Each neuro-symbolic decision retains the facts that informed it, the model and rule versions used, the reasoning trace, any changes to the risk score, human-review status, and the eventual outcome.

For example, a risk-adjustment record would show the original score, the symbolic and neural adjustments, and the final score, together with a short explanation of why the change occurred. This might include an anomaly score crossing a defined threshold, similarity to previous true-positive incidents, and the absence of a matching benign suppression. Keeping this information with the decision makes the influence of neural and symbolic components traceable, reproducible, and, where necessary, reversible.

12. Evaluation Framework

NS-D&R can be evaluated across three areas. Detection quality measures whether the system improves alert fidelity, including true- and false-positive rates, precision across cluster types, time to triage and resolution, and the accuracy of suppression decisions. Neuro-symbolic quality focuses on the additional reasoning introduced by the architecture, including the usefulness of explanations, analyst agreement with recommendations, the accuracy of risk adjustments, similar-alert retrieval and causal links, and model calibration.

Operational quality considers the cost of introducing these capabilities into a production SOC, including additional pipeline latency, fallback behaviour when models are unavailable, processing cost, and the number of recommendations accepted or rejected by analysts. Taken together, these measures assess not only whether NS-D&R improves detection, but whether those improvements remain explainable, useful to analysts, and practical to operate.

13. Discussion

The architecture deliberately places different levels of trust in its neural and symbolic components. Neural models are given considerable freedom to propose relationships, identify patterns, and generate hypotheses, but much less freedom to make production decisions. This makes neural capability more practical in environments where alerting needs to remain predictable and explainable.

Analyst decisions and closure outcomes also provide a continuing source of feedback for both neural models and symbolic rules. This allows the system to adapt over time while keeping human judgement at the centre of the process. Just as importantly, neural components remain an augmentation to the deterministic pipeline. If a model becomes unavailable or unreliable, its contribution can be removed and the system can fall back to the existing detection baseline without interrupting alerting.

These safeguards also impose trade-offs. Bounding neural influence may prevent the system from acting fully on patterns that learned models identify correctly, while retaining deterministic controls preserves some of the limitations of the existing detection pipeline. The architecture also depends on the quality of its underlying telemetry, entity resolution, historical outcomes, and analyst feedback; errors or bias in these inputs can affect both learned representations and subsequent symbolic rules. NS-D&R therefore does not eliminate uncertainty from detection and response, but constrains where that uncertainty is permitted to influence operational decisions.

14. Limitations and Future Work

The architecture has several limitations. Its effectiveness depends partly on the availability and quality of historical outcomes. In environments with limited labelled data, similarity and calibration will be less reliable until enough analyst decisions and closure outcomes have accumulated. Learned representations may also reproduce patterns or biases in that history, potentially making genuinely novel attacks appear more benign than they are. Bounded neural influence and symbolic controls reduce this risk but cannot remove it entirely.

The response assistant introduces a different concern. Even when grounded in available evidence, an LLM can still produce convincing but incorrect hypotheses. Supporting and contradicting evidence should therefore remain visible to analysts, with consequential actions requiring human approval. The limits placed on neural influence also involve a trade-off between sensitivity and safety, and the appropriate bounds are likely to differ between environments.

Future work should examine adaptive bounds based on measured model calibration, formal validation of symbolic checks over proposed causal relationships, and controlled studies of how neuro-symbolic assistance affects analyst trust, workload, and decision quality.

15. Conclusion

This paper presents a neuro-symbolic detection and response architecture designed to extend, rather than replace, an existing signal-based detection pipeline. Neural models are used for tasks where learning and generalisation are valuable—discovering patterns, building representations, ranking evidence, summarising activity, and generating hypotheses—while symbolic components retain responsibility for validation, constraints, scoring, explanation, and governance.

The central principle is that neural capability does not have to come at the expense of operational control. By bounding its influence on production decisions and retaining the deterministic pipeline as the underlying authority, NS-D&R can introduce learned behavioural and cross-entity context while preserving explainability, auditability, and human oversight. The three-plane architecture, detection knowledge graph, bounded calibration model, adoption roadmap, and evaluation framework together provide a vendor-neutral approach for introducing neuro-symbolic capabilities into security operations.

References

Apruzzese G., Laskov P., Schneider J. (2023). SoK: Pragmatic assessment of machine learning for network intrusion detection. In *2023 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE.
https://doi.org/10.1109/EuroSP57164.2023.00042

Aspis Y., Broda K., Lobo J., Russo A. (2022). Embed2Sym: Scalable neuro-symbolic reasoning via clustered embeddings. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning* (pp. 421–431).
https://doi.org/10.24963/kr.2022/41

Badreddine S., d'Avila Garcez A. S., Serafini L., Spranger M. (2022). Logic Tensor Networks. *Artificial Intelligence*, 303, 103649.
https://doi.org/10.1016/j.artint.2021.103649

Barbiero, P., Ciravegna, G., Giannini, F., Espinosa Zarlenga, M., Magister, L. C., Tonda, A., Liò, P., Precioso, F., Jamnik, M., & Marra, G. (2023). Interpretable neural-symbolic concept reasoning. In *Proceedings of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research, 202*, 1801–1825.

https://proceedings.mlr.press/v202/barbiero23a.html

Besold T. R., d'Avila Garcez A., Bader S., Bowman H., Domingos P., Hitzler P., Kühnberger K.-U., Lamb L. C., Lowd D., Lima P. M. V., de Penning L., Pinkas G., Poon H., Zaverucha G. (2017). Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.
https://arxiv.org/abs/1711.03902

Cheng Z., Lv Q., Liang J., Wang Y., Sun D., Pasquier T., Han X. (2024). KAIROS: Practical intrusion detection and investigation using whole-system provenance. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE.
https://doi.org/10.1109/SP54263.2024.00110

d'Avila Garcez A. S., Gori M., Lamb L. C., Serafini L., Spranger M., Tran S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *IfCoLog Journal of Logics and their Applications*, 6(4), 611–632.
https://arxiv.org/abs/1905.06088

Eckhoff M. W., Halvorsen J., Hansen B. J., Eian M., Mavroeidis V., Chetwyn R. A., Skjøtskift G., Grov G. (2025). Experimenting with neurosymbolic artificial intelligence for defending against cyber attacks. *Neurosymbolic Artificial Intelligence*.
https://doi.org/10.1177/29498732251377352

Garnelo M., Shanahan M. (2019). Reconciling deep learning with symbolic artificial intelligence: Representing objects and relations. *Current Opinion in Behavioral Sciences*, 29, 17–23.
https://doi.org/10.1016/j.cobeha.2018.12.010

Guo C., Pleiss G., Sun Y., Weinberger K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research* (Vol. 70, pp. 1321–1330). PMLR. https://proceedings.mlr.press/v70/guo17a.html

Hamilton W. L., Ying R., Leskovec J. (2017). Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3), 52–74. https://arxiv.org/abs/1709.05584

Han X., Pasquier T., Bates A., Mickens J., Seltzer M. (2020). UNICORN: Runtime provenance-based detector for advanced persistent threats. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2020.24046

Hassan W. U., Bates A., Marino D. (2020). Tactical provenance analysis for endpoint detection and response systems. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 1172–1189). IEEE. https://doi.org/10.1109/SP40000.2020.00096

Hitzler P., Eberhart A., Ebrahimi M., Sarker M. K., Zhou L. (2022). Neuro-symbolic approaches in artificial intelligence. *National Science Review*, 9(6), nwac035. https://doi.org/10.1093/nsr/nwac035

Liu F. T., Ting K. M., Zhou Z.-H. (2008). Isolation Forest. In 2008 Eighth IEEE International Conference on Data Mining (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17

Manhaeve R., Dumancic S., Kimmig A., Demeester T., De Raedt L. (2018). DeepProbLog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems* (Vol. 31, pp. 3749–3759). https://proceedings.neurips.cc/paper/2018/hash/3e77a14629775492504515dc4b23a894-Abstract.html

Ribeiro M. T., Singh S., Guestrin C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). ACM. https://doi.org/10.1145/2939672.2939778

Tabassi E. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). *NIST AI 100-1*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

Yang Z., Ishay A., Lee J. (2020). NeurASP: Embracing neural networks into Answer Set Programming. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (pp. 1755–1762). https://doi.org/10.24963/ijcai.2020/243

Zhou C., Paffenroth R. C. (2017). Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 665–674). ACM.

https://doi.org/10.1145/3097983.3098052

Zhou A., Xu X., Raghunathan R., Lal A., Guan X., Yu B., Li B. (2024). KnowGraph: Knowledge-enabled anomaly detection via logical reasoning on graph data. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security* (pp. 168–182). ACM.
https://doi.org/10.1145/3658644.3690354

Appendix A.

Illustrative Case Study: Vulnerability-to-Detection Lifecycle

Consider what happens when a new vulnerability appears in a CVE feed, vendor bulletin, external advisory, or threat-intelligence report. Before a detection can be written, someone in the SOC has to answer a more basic question: does this vulnerability actually matter here? That usually means checking whether the affected technology is present, working out what telemetry could expose exploitation, developing and testing detection logic, and deciding whether the resulting rule is reliable enough to deploy. NS-D&R does not remove these steps. Instead, it uses neural components to assist with the parts that involve interpretation and comparison, while symbolic controls remain responsible for deciding what is relevant and what can enter production.

The advisory first needs to be turned into information the detection pipeline can use. This might include the affected product and versions, platform, vulnerability class, exploit conditions, known indicators, and any attacker behaviours or tactics described in the report. A neural model can help interpret this material and look for useful precedents in earlier vulnerabilities, detections, and incidents. A newly disclosed deserialization flaw, for instance, may have little in common with an older vulnerability at the software level but still lead to familiar behaviour at the host: an unusual child process, an unexpected outbound connection, or a file being written after an application request. Those similarities may be more useful to a detection engineer than the vulnerability description alone.

Whether they matter, however, depends on the environment. This is where the symbolic side of the architecture becomes important. The detection knowledge graph can connect the advisory to relevant assets and cloud resources, together with contextual information about software dependencies, vulnerabilities, exposure, and endpoint coverage. A serious FreeBSD vulnerability is of limited immediate concern to an organisation that has no FreeBSD systems or dependent products. The reverse can also be true. A vulnerability with a less alarming severity rating may deserve urgent attention if it affects an identity service used across the organisation or is exposed to the Internet on critical systems. The distinction is between what is generally dangerous and what is relevant to this particular environment.

These different signals inform the review of whether a candidate detection should be developed or progressed. Some come from neural components: similarity to previously exploited vulnerabilities, confidence in a candidate detection, or uncertainty about how the advisory has been interpreted. Others are concrete organisational facts, such as whether affected systems exist, whether they are exposed, whether the required logs are collected, and

how noisy the proposed detection is expected to be. The neural assessment is useful here, but it is not decisive. A strong resemblance to an earlier exploited vulnerability does not help much if the affected software is absent, nor can a proposed rule be deployed if the telemetry needed to support it is unavailable.

For straightforward cases, the system can use an approved template to draft a candidate detection and send it through the normal detection pipeline. It still carries its provenance with it: the source advisory, affected assets, required telemetry, test results, rationale, and rollback information. The rule might also spend some time in shadow or canary mode before it begins producing operational alerts. Existing controls remain in place during this process. Field references must be valid, expected alert volume needs to be reasonable, and severity and deduplication behaviour must be defined. Promotion to production remains subject to the organisation's approval process.

Not every case should move that quickly. If the review identifies greater uncertainty or risk, the system instead hands the case to a detection engineer. Rather than presenting only a generated rule, it can show why the case needs attention: which systems appear to be exposed, which earlier detections look similar, what evidence supports the proposed logic, and where the gaps are. An organisation might, for example, run the vulnerable software on several Internet-facing systems but have incomplete telemetry for the behaviour the proposed rule relies on. That is a useful recommendation, but not a good candidate for automatic deployment. The engineer can modify the rule, reject it, ask for more evidence, or approve it with appropriate controls.

The example also shows why NS-D&R keeps neural and symbolic responsibilities separate. Neural models can shorten the path from an advisory to something a detection engineer can investigate or test. They are useful for finding similarities that are difficult to express as fixed rules and for producing a reasonable starting point for new detection logic. They do not decide whether that logic belongs in production. That decision still depends on what is known about the environment, whether the evidence supports the rule, and whether the organisation's production controls have been satisfied.

Appendix B — Formal Definitions and Architectural Constraints

This appendix provides additional definitions for the formal expressions introduced in Sections 4, 6.5, and 6.6. The equations specify architectural boundaries and decision constraints within NS-D&R rather than a particular learning algorithm, model family, or optimisation method. Their purpose is to make explicit how neural outputs enter the architecture, how their influence is bounded, and how proposed neural relationships become operational facts.

B.1 Neural Context and Decision Authority

Section 4 defines the separation between neural inference and production authority as:

$C_n = N(x)$ (1)

$D = S(x, C_n)$ (2)

Here, x represents the evidence available for a finding, N the neural components, C_n the context produced by those components, S the symbolic reasoning layer, and D the resulting production decision.

Equation (1) allows neural components to derive additional context from the available evidence. This context may include anomaly scores, embeddings, similarity results, candidate relationships, or hypotheses. Equation (2) places the resulting production decision within the symbolic layer, which can consider both the original evidence and the neural context.

The neural output therefore influences production decisions only through the symbolic layer. It has no independent authority to determine D. Instead, S may accept, constrain, modify, or discard its contribution according to the applicable rules and policy.

It has no independent authority to determine production decisions; the symbolic layer may accept, constrain, modify, or discard its contribution according to the applicable rules and policy.

B.2 Bounded Risk Adjustment

Section 6.5 defines the neural contribution to risk as:

$$\hat{\Delta}_n = clip(\Delta_{n,} - \lambda, \lambda) \qquad (3)$$

The final risk score is then:

$$R_f = clip(R_b + \Delta_s + \Delta_h + \Delta_c - \Delta_b + \hat{\Delta}_{n,} R_{min,} R_{max}) \qquad (4)$$

subject to the safety invariants:

$$Suppress(A, N) = 0 \qquad (5)$$

$$\text{Severity}(A, N) \neq S_{max} \qquad (6)$$

The notation is defined as follows:

Symbol	Definition
R_b	Human-authored base risk score
R_f	Final production risk score
Δ_s	Adjustment produced by deterministic symbolic risk rules
Δ_h	Historical-fidelity adjustment derived from prior outcomes
Δ_c	Adjustment associated with a symbolically validated causal chain
Δ_b	Reduction associated with deterministic known-benign context
Δ_n	Neural-context risk adjustment before bounding
$\hat{\Delta}_n$	Neural-context adjustment after bounding
λ	Maximum permitted magnitude of the neural contribution
R_{min}	Minimum permitted production risk score
R_{max}	Maximum permitted production risk score
S_{max}	Highest production severity
A	An otherwise alertable finding or alert
N	Neural evidence considered without independent symbolic justification

Table B.1. Notation for the Bounded Risk Adjustment Model

Equation (3) limits the neural contribution to the interval [−λ, λ]. Equation (4) then combines this bounded contribution with the base score and the permitted symbolic adjustments. The outer clipping operation ensures that the resulting value remains within the production scoring range.

For example, consider a finding with R_b = 60, Δ_s= 5, Δ_h = 3, Δ_c = 5, and Δ_b = 0. Suppose the neural components propose Δ_n = 18, while policy sets λ = 10. Equation (3) gives:

$$\hat{\Delta}_n = clip(18, -10, 10) = 10$$

Assuming a production range of [0, 100], Equation (4) then gives:

$$R_f = clip(60 + 5 + 3 + 5 - 0 + 10, 0, 100) = 83$$

Without the bound, the neural contribution would have produced a score of 91. The example illustrates the intended role of neural evidence: it can materially affect a borderline decision, but the architecture places an explicit limit on that influence.

Equations (5) and (6) impose additional constraints that are not captured by score bounding alone. Neural evidence by itself cannot suppress an otherwise alertable finding and cannot independently assign the highest production severity. These invariants apply even where the bounded score produced by Equation (4) would otherwise cross the corresponding production threshold.

B.3 Neural Edge Proposal and Symbolic Validation

Section 6.6 formalises the relationship between neural clustering and symbolic validation.

For two findings f_i and f_j, a neural component may propose the candidate edge:

$$e^N_{ij} = (f_i, f_j, c_{ij}) \qquad (7)$$

where $c_{ij} \in [0,1]$ represents the neural relationship score assigned to the proposed edge, with higher values indicating stronger learned evidence that the two findings are related.

The candidate is then evaluated by the symbolic validator:

$$V(e^N_{ij}) = T_{ij} \wedge E_{ij} \wedge A_{ij} \wedge L_{ij} \quad (8)$$

where:

Symbol	Definition
f_i, f_j	Findings being evaluated for a relationship
e^N_{ij}	Relationship proposed by a neural component
c_{ij}	Neural relationship score for the proposed relationship
T_{ij}	Temporal compatibility between the findings
E_{ij}	Shared or linked entity relationship
A_{ij}	Compatibility between associated adversary tactics or techniques
L_{ij}	Evidence continuity across shared contextual evidence
V	Symbolic validation function
e^*_{ij}	Accepted operational edge
$\emptyset$	No operational edge is established between the findings

Table B.2. Notation for Neural Edge Proposal and Symbolic Validation

The operational edge is defined as:

$$e_{ij}^{*} = \begin{cases} e_{ij}^{N} & \text{if } \mathrm{V}\left(e_{ij}^{N}\right) = 1 \\ \emptyset & \text{if } \mathrm{V}\left(e_{ij}^{N}\right) = 0 \end{cases} \qquad (9)$$

Here, $\emptyset$ does not assert that the findings are unrelated. It indicates that the proposed relationship failed the applicable symbolic validation and is therefore not admitted as an operational edge for clustering or downstream reasoning. The neural proposal may still be retained as contextual information for analyst review, model evaluation, or subsequent analysis.

The relationship score c_{ij} is produced by the neural relationship component from learned representations of the two findings and their surrounding context. Depending on the implementation, this may incorporate behavioural and semantic similarity, entity representations, temporal patterns, and similarity to historical finding pairs that were previously grouped into the same incident. NS-D&R does not prescribe a particular model for producing this score; c_{ij} provides a common interface between the neural proposal stage and the symbolic validator.

Consider a neural component that proposes a relationship with a neural relationship score of $c_{ij} = 0.94$. The symbolic checks return:

$T_{ij} = 1$
$E_{ij} = 1$
$A_{ij} = 0$
$L_{ij} = 1$

Equation (8) therefore gives:

$$V\left(e_{ij}^{N}\right) = 1 \wedge 1 \wedge 0 \wedge 1 = 0$$

and Equation (9) consequently gives:

$e_{ij}^{*}=\emptyset$

The candidate relationship is rejected despite the neural component assigning it a relationship score of 0.94. The example illustrates the distinction between a neural relationship score and operational authority, the score represents the strength of the learned evidence for a proposed relationship, while the symbolic constraints determine whether that relationship can become part of the operational clustering state.

Together, Equations (1)–(9) formalise the three constraints that underpin NS-D&R, neural outputs enter the system as context rather than decisions, their influence on production scoring is explicitly bounded, and neural relationships must satisfy symbolic constraints before becoming operational facts.

Appendix C — Diagram Source Definitions

The figures used throughout this article were produced from Mermaid diagram definitions. The source definitions are reproduced below to make the architectural relationships explicit and to support reproduction or adaptation of the figures. Styling distinguishes neural components from symbolic components where this distinction is relevant to the architecture.

C.1 NS-D&R Architecture

The following definition corresponds to the overall NS-D&R architecture, showing the path from normalized telemetry through the existing detection pipeline, neuro-symbolic context layer, bounded risk adjustment, clustering, alerting, and analyst-facing response.

```
flowchart TB

T[Telemetry]:::neutral --> N[Schema normalization]:::neutral

N --> D["Detection analytics: symbolic, authoritative"]:::sym

D --> F[Structured findings]:::neutral

subgraph NSL[Neuro-symbolic context layer]

direction LR

EMB[Neural embedding and anomaly]:::neural

HYP[Hypothesis generator]:::neural

KG[Detection knowledge graph]:::sym

POL[Symbolic policy and rule engine]:::sym

CAU[Causal attack-path reasoner]:::sym

end

F --> NSL

NSL --> RA["Bounded risk adjustment: symbolic plus capped neural delta"]:::sym

RA --> P[(Persistence: findings plus NS context)]

P --> CL[Neuro-symbolic clustering]

P --> DC[Deterministic clustering]:::sym
```

```
CL --> AL["Alerting stage: production control point"]:::sym

DC --> AL

AL --> RESP["Response assistant: human-approved"]:::neural

RESP -. closure outcomes and tuning .-> EMB

RESP -. rule changes .-> POL

classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;

classDef sym fill:#E3FCEF,stroke:#006644,color:#172b4d;

classDef neutral fill:#EFEFEF,stroke:#4a4a4a,color:#172b4d;
```

C.2 Detection Knowledge Graph

This definition illustrates the principal node and relationship types in the detection knowledge graph. It is intended as a representative schema rather than an exhaustive graph model.

```
flowchart LR

DET[Detection] -->|produces| FIND[Finding]

FIND -->|involves| USER[User]

FIND -->|observed on| DEV[Device]

FIND -->|observed on| CLOUD[Cloud resource]

FIND -->|has evidence| PROC[Process]

FIND -->|maps to| TECH[Adversary technique]

RULE[Risk rule] -->|adjusts| FIND

FIND -->|included in| CLU[Alert cluster]

CLU -->|generates| ALERT[Alert]

ALERT -->|triaged as| OUT[Closure outcome]

PLAY[Playbook] -->|applies to| CLU

TUNE[Tuning ticket] -->|modifies| DET

OUT -.->|calibrates| RULE
```

C.3 Entity Embeddings and Behavioural Anomaly

The following definition shows how rolling entity activity is transformed into learned representations and associated anomaly and similarity signals before being stored as neuro-symbolic context.

```
flowchart LR

    EV[Entity events: rolling window] --> FE[Feature extraction]

    FE --> ENC[Neural encoder: sequence / autoencoder / contrastive]:::neural

    ENC --> EMB[(Entity embedding store)]

    EMB --> AN[Self-anomaly score]:::neural

    EMB --> PEER[Peer-group deviation]:::neural

    EMB --> SIM[Nearest historical clusters + labels]:::neural

    AN --> CTX[Neuro-symbolic context record]

    PEER --> CTX

    SIM --> CTX

    classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;
```

C.4 Neuro-Symbolic Enrichment

This definition shows the construction of the neuro-symbolic context record and the architectural constraint that prevents neural enrichment from directly emitting an alert.

```
flowchart LR

    FIND[Finding] --> ENR[Enrichment stage]

    AN[Anomaly score]:::neural --> ENR

    SIM[TP / FP similarity]:::neural --> ENR

    HYP[Attack-stage hypotheses]:::neural --> ENR

    ENR --> NSCTX[Neuro-symbolic context record]
```

```
    NSCTX --> SCORE[Symbolic scoring]:::sym

    NSCTX --> CLUST[Symbolic clustering]:::sym

    NSCTX --> RESP[Response assistant]:::neural

    NSCTX -. never directly emits .-x ALERT[Alert]

    classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;

    classDef sym fill:#E3FCEF,stroke:#006644,color:#172b4d;
```

C.5 Symbolic Risk Reasoner

The symbolic risk reasoner consumes neural context alongside deterministic facts. Neural evidence can therefore contribute to a risk decision without independently determining that decision.

```
flowchart TB

    NSCTX[Neuro-symbolic context]:::neural --> RR{Neuro-symbolic risk rule}:::sym

    BEN[Known-benign / suppression facts]:::sym --> RR

    RR -->|anomaly high AND TP-similar AND approved detection| BOOST[Confidence
boost]

    RR -->|resembles benign pattern| DAMP[FP dampener]

    RR -->|coherent attack-stage chain| CHAIN[Causal-chain boost]

    RR -->|deterministic benign context| OVER[Symbolic override / suppress]

    BOOST --> DELTA[Deterministic, explainable risk delta]

    DAMP --> DELTA

    CHAIN --> DELTA

    OVER --> DELTA

    classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;

    classDef sym fill:#E3FCEF,stroke:#006644,color:#172b4d;
```

C.6 Bounded Confidence Calibration

This definition corresponds to the bounded risk model formalised in Equations (3)–(6). The neural-context delta is constrained before entering the final score, after which the safety invariants governing suppression and maximum severity are applied.

```
flowchart LR

BASE[Human-authored base score]:::sym --> SUM(( + ))

SYMD[Symbolic rule delta]:::sym --> SUM

HIST[Historical-fidelity delta]:::sym --> SUM

CAUS[Validated causal-chain delta]:::sym --> SUM

BEN[Known-benign reduction]:::sym --> SUM

NEUR[Neural-context delta]:::neural --> CAP[Clamp to bounded range]:::sym

CAP --> SUM

SUM --> INV{Safety invariants}:::sym

INV -->|neural alone cannot suppress<br/>or set top severity| FINAL[Final risk
score]

classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;

classDef sym fill:#E3FCEF,stroke:#006644,color:#172b4d;
```

C.7 Neuro-Symbolic Clustering

The clustering definition shows the deterministic clustering baseline alongside neural-assisted semantic, causal, and multi-entity proposals. Candidate relationships are subject to symbolic validation before they can contribute to the clustered findings supplied to alerting.

```
flowchart TB

    FIND[Findings] --> DC[Deterministic clustering: shared keys]:::sym

    FIND --> SEM[Semantic expansion: embeddings]:::neural

    FIND --> CC[Causal-edge proposal]:::neural
```

```
    FIND --> ME[Multi-entity grouping]:::neural

    CC --> VAL{Symbolic validation:<br/>time order, shared entities, tactic
fit}:::sym

    SEM --> VAL

    ME --> VAL

    VAL --> MERGE[Validated clustered findings]

    DC --> MERGE

    MERGE --> AL[Existing alerting stage]:::sym

    classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;

    classDef sym fill:#E3FCEF,stroke:#006644,color:#172b4d;
```

C.8 Response Plane: Neuro-Symbolic Triage Assistant

The final definition shows the analyst-facing response assistant. The assistant combines alert and finding data with knowledge-graph facts, historical outcomes, and playbooks to produce a grounded explanation for human review.

```
flowchart LR

    CLU[Alert cluster + findings]:::sym --> ASST[Triage assistant: LLM +
retrieval]:::neural

    KG[Knowledge-graph facts]:::sym --> ASST

    HIST[Historical outcomes]:::sym --> ASST

    PLAY[Playbooks]:::sym --> ASST

    ASST --> EXPL["Grounded explanation: hypothesis, supporting/contradicting
facts, recommended playbook + pivots"]

    EXPL --> ANALYST{Human analyst}:::sym

    ANALYST -. feedback / closure .-> HIST

    classDef neural fill:#EEF0FF,stroke:#5B5BD6,color:#172b4d;

    classDef sym fill:#E3FCEF,stroke:#006644,color:#172b4d;
```