The Eastasouth Journal of Information System and Computer Science
Vol. 1, No. 01, August, pp. 241 - 252

ThreatWeave-X: A Systematic Review of Graph Neural Learning for Predicting Latent Multi-Stage Cyberattack Paths from Fragmented Enterprise Security Signals

Sujatha Lakshmi Narra
Robinhood Markets, Cumming, GA, USA

Article Info

Article history:

Received Aug, 2023
Revised Aug, 2023
Accepted Aug, 2023

Keywords:

Advanced Persistent Threats;
Attack-Graph Generation;
Attention Mechanisms;
Enterprise Threat Intelligence;
Graph Neural Networks;
Heterogeneous Graph Learning;
Multi-Stage Attack Detection;
Provenance-Graph Analytics

ABSTRACT

The telemetry being used by enterprise security operations is increasingly endpoint, network, identity and cloud-based, with each individually exposing only part of an adversary's actions, and multi-stage intrusions, like those stemming from advanced persistent threats, very hard to reconstruct from any single source. We present a systematic review of the methodological heritage from deterministic attack-graph generation to provenance-graph and log-based intrusion analytics, and graph neural network representation learning that spans twenty-six papers published over the past 20 years (2002-2022). It chronicles the journey from the relational and infrastructure-aware formulations of attack graph research to the inductive, attentional and relation-aware formulations of graph neural architectures, that emerged as a direct result of their scalability challenges, and the evolution of provenance-graph analytics from a purely sequence-based treatment of system behavior to a more explicit causal and relational treatment. On the basis of this synthesis, the review suggests a five-stage conceptual framework called ThreatWeave-X, which integrates multi-source signals from various sources into a single heterogeneous provenance graph, relational and attentional graph-neural embedding and a sequence-aware attention mechanism with an analyst feedback loop to rank candidate multi-stage attack paths. The review also outlines four cross-cutting challenges that appear in the reviewed literature that are still not completely addressed: 1) incompleteness of multi-modal heterogeneity handling, 2) scarcity of labels, 3) limited interpretability, and 4) inconsistent evaluation protocols, which all suggest directions for future empirical validation of the graph-neural multi-stage attack-path prediction.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Name: Sujatha Lakshmi Narra
Institution: Robinhood Markets, Cumming, GA, USA
Email: SujathaNarra87@gmail.com

1. INTRODUCTION

Telemetry from an endpoint detection and response (EDR) agent, network flow monitors, identity and access management (IAM) systems and cloud audit trails are now vital to enterprise security operations. Each of these streams captures only a portion of the adversary's actions; a log of suspicious process launch, a record of an unusual authentication event, or an unusual outbound connection is not necessarily

enough to confirm an adversary has compromised the system. Advanced persistent threats (APTs) rely on this very fragmentation, with reconnaissance, lateral movement, privilege escalation, and exfiltration taking place over a long period of time and numerous hosts, and no single alert ever revealing the entire attack story [1]. This presents the analysts with an alert-correlation challenge: the fragmented, weakly labeled high volume of alerts needs to be correlated together in order to identify a multi-stage attack path which can then be prioritized for a defensive response.

There are two complementary research traditions that have been tackled to solve this problem. The first creates deterministic attacks graphs that count the reachability of adversary between vulnerable configurations and provide formal guarantees, based on pre-enumerated exploit and topology models. The second addresses the processing of logs in the system and network and uses statistical, sequence-based or graph-analytical techniques to detect, via the logs, the presence of substructures that are anomalous. Recently, both traditions have been focused on graph representation learning, where entities and their relationships are represented in continuous vector spaces that can be leveraged by a downstream classifier or ranking function [2]. The convergence impels the present review.

The overall goal of this systematic review is threefold: (1) synthesize the methodological lineage between classical attack-graph generation, provenance-graph and log based intrusion analytics, and graph neural network (GNN) architectures; (2) identify common design choices, datasets and evaluation practices found in this literature up to the review's coverage horizon, and (3) propose a conceptual synthesis framework, dubbed ThreatWeave-X, which brings together these design choices into a consistent processing pipeline to predict threat latent multi-stage attack paths from fragmented enterprise signals. The framework is presented in a conceptual and architectural way, rather than as an empirically validated system, which corresponds to the synthesis-oriented character of the reviewed literature [3].

The rest of this paper is organized as follows. Section 2 surveys the literature under four themes, which are attack-graph foundations, attack-graph and log-based detection, graph representation learning and neural architectures, and benchmark datasets. The methodology for the review is described in Section 3. Section 4 introduces the ThreatWeave-X synthesis framework, formalization, and discusses cross-cutting challenges. The paper ends with section 5 [4].

2. LITERATURE REVIEW

2.1 Attack-Graph Foundations

The formal generation of attack-graphs was introduced by model-checking based methodology, where every state of a network is a node and pre-conditions (such as user actions or events) are edges, and exhaustively generate adversary reachability towards a defined goal state. While expressive, exhaustive state enumeration is a poor scaling function with number of hosts and vulnerabilities, the motivation for relational and logic-programming reformulations of attack graphs is that they express attack graphs as derivations from host, service, and vulnerability predicates, which leads to much better scalability to enterprise-sized networks. Following this study, a reworking of attack-graph construction was released as a polynomial time, monotonic algorithm based on the dependencies between exploits, thereby relieving the computational load caused by large topologies. To keep static graph models up-to-date with the emerging enterprise infrastructure, attack-graph formalisms are extended to represent new communication semantics for multiple protocols and current information-technology architectures like cloud and containerized environments [5]. One of the key challenges in this whole line of research is the need to depend on manually curated vulnerability and topology databases, which are expensive

to maintain and incapable of detecting attacks that don't follow the vulnerability and topology sets that have already been enumerated. We place this deterministic strand in comparison to the ones discussed in Sections 2.2 and 2.3, which are provenance-graph and graph-neural based respectively. This deterministic strand is positioned in relation to the provenance-graph and graph-neural based deterministic strands discussed in Sections 2.2 and 2.3 respectively.

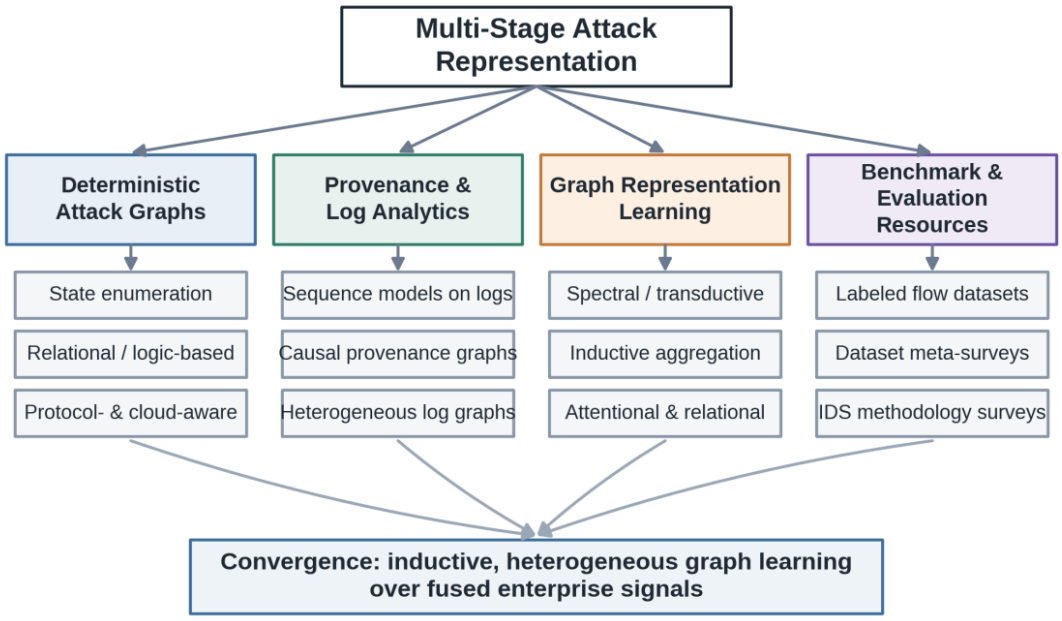


Figure 1. Taxonomy of Multi-Stage Attack Representation Approaches
Branches converge toward inductive, heterogeneous graph learning over fused enterprise signals.

Table 1. Chronological Overview of Attack-Graph Modeling Approaches

Study	Year	Graph Formalism	Core Mechanism	Scalability Strategy
[6]	2002	State-enumeration attack graph	Model-checking over network states	Exhaustive search; limited scalability
[2]	2002	Relational dependency graph	Predicate-based exploit chaining	Polynomial-time relational encoding
[7]	2006	Logic-based attack graph	Monotonic exploit-dependency reasoning	Polynomial-time generation
[8]	2022	Extended protocol-aware attack graph	Multi-protocol, cloud/IT-aware modeling	Modular extension to modern architectures

Note. Entries are ordered chronologically to illustrate the trajectory from exhaustive enumeration toward scalable, infrastructure-aware formulations.

2.2 *Provenance-Graph and Log-Based Detection*

The second strand of research is to build the graph directly from the operational telemetry, without relying on vulnerability models. The recent approach of treating system logs as structured sequences of events was shown to be effective in detecting deviations from normal execution sequences by DeepLog, which was able to train a recurrent sequence model on these normal sequences to identify deviations due to unusual execution paths, but did not retain the relational structure of processes, files and sockets. To overcome this, Provenance-graph approaches explicitly represent causal

relationships between system entities. HOLMES integrates low-level audit events into high-level narratives of attack stages, based on a kill-chain model, to reduce the burden of the analysts to suppress benign causal chains. POIROT uses inexact graph-pattern matching to search for known adversary techniques by matching threat-intelligence derived query graphs with kernel audit provenance graphs, that is, looking back in history for known adversary techniques. A similar cause-and-effect analysis strategy emphasizes suspicious event chains by rolling the scores of anomalies backward and forward in a provenance graph to provide analysts with a timely way to investigate causal chains of true-positive events. Log2vec builds a heterogeneous graph from authentication and process-execution logs, and uses graph-embedding algorithms to identify insider and lateral-movement threats, without the need to hand-crafted rules. The concept has also been generalized in a variety of streaming forms, including the detection of anomalies in a heterogeneous graph while it is still streaming without materializing it in memory [9]. In this strand, the completeness and fidelity of collecting audit logs has been shown to be a primary factor affecting detection quality; and most systems are still host-centric, and require extra engineering to join network- and identity-layer signals, as described in Table 2.

Table 2. Provenance-Graph and Log-Based Enterprise Detection Systems

System	Primary Data Source	Graph / Sequence Unit	Core Detection Principle
DeepLog [4]	System execution logs	Event sequence	Sequence-model deviation detection
Causality analysis [10]	Kernel audit logs	Causal provenance graph	Backward/forward suspicion propagation
Streaming graph detection [11]	Streaming heterogeneous graphs	Streaming graph sketch	Memory-efficient anomaly scoring
HOLMES [12]	Kernel audit records	Correlated provenance subgraph	Kill-chain-stage correlation
POIROT [13]	Kernel audit + threat intelligence	Query-to-provenance graph match	Inexact graph-pattern threat hunting
Log2vec [14]	Authentication and process logs	Heterogeneous log graph	Graph-embedding insider/lateral-movement detection
Threatrace [15]	System provenance graph	Node-level embedding	Inductive node-level anomaly tracing

Note. Systems are drawn from the reviewed literature and are listed in the order discussed in Section 2.2.

2.3 *Graph Representation Learning and Graph Neural Networks*

The original graph neural network formulation paved the way for learning directly on graph-structured data by iteratively propagating states around the graph until convergence for all graphs of arbitrary topology. This was later rephrased in a localized approximation of spectral graph convolutions (LGC) by spectral graph convolutional networks, which was shown to be an efficient, layer-wise propagation rule for semi-supervised node classification. Early approaches were transductive, meaning that the entire graph is needed during training, so GraphSAGE added an inductive approach that learns aggregator functions over sampled neighbourhoods to allow embedding of unseen nodes and to make large, dynamically evolving enterprise graphs tractable. In addition to the convolutional aggregation, graph attention networks introduced a new mechanism called attention, which

learned the weights for the aggregation of neighboring entities based on their relevance to the target entity, instead of the raw connectivity. Relational graph convolutional networks adapted the convolutional aggregation by adding relation-specific transformation matrices to the graph in order to prevent the fusion of different semantic types of edges between nodes, and heterogeneous graph attention networks proposed hierarchical node-level and semantic-level attention that incorporates multiple metapaths to model local structure and higher-order semantic relevance, respectively. However, previous shallow embedding approaches, especially node2vec, have demonstrated that the embeddings can be homophilous, meaning that they capture community structure, or structural-equivalence roles based on their walk parameters; these types of embeddings have been used to guide the design of walk-based sampling strategies in subsequent graph-neural pipelines. In a direct convergence of the strands mentioned in Sections 2.2 and 2.3, Threatrace took the inductive, node-level graph representation learning approach directly to system provenance graphs, tracing anomalous entities through the graph without a pre-defined attack signature. The survey overall verifies that convolutional, attentional and recurrent message passing designs are a well-developed design space, and that there are recognised open challenges with regard to scalability, dynamic graph learning and interpretability [16]. The main architectures mentioned in this subsection are summarized in Table 3.

Table 3. Graph Representation Learning and Graph Neural Network Architectures Reviewed

Method	Year	Learning Paradigm	Aggregation Mechanism	Primary Contribution
node2vec [5]	2016	Shallow embedding	Biased random-walk sampling	Flexible homophily / structural-equivalence trade-off
GNN (original) [17]	2009	Recursive propagation	Iterative fixed-point propagation	Formal foundation for learning on graphs
GCN [18]	2017	Transductive	Localized spectral convolution	Efficient semi-supervised node classification
GraphSAGE [9]	2017	Inductive	Sampled-neighborhood aggregation	Generalization to unseen nodes / large graphs
GAT [19]	2018	Inductive / transductive	Learned attention-weighted aggregation	Relevance-weighted neighbor aggregation
R-GCN [20]	2018	Transductive / inductive	Relation-specific weight matrices	Multi-relational heterogeneous modeling
HAN [21]	2019	Inductive	Node- and semantic-level attention	Metapath-based heterogeneous representation

Note. Learning paradigm reflects whether the method requires the full graph during training (transductive) or generalizes to unseen nodes (inductive).

2.4 *Benchmark Datasets, Detection Baselines, and Evaluation Practices*

In this literature, a relatively limited set of common resources have been used to evaluate the empirical results. The UNSW-NB15 dataset was synthesized to overcome the limitations of previous intrusion databases, which consist of old attack sets with limited diversity of attack types, and has been used as a standard reference set of attacks and background traffic for both traditional and learning-based network intrusion detection. A survey of network-based intrusion detection resources identified the benefits and challenges of these resources, and observed that class imbalance,

insufficient ground truth documentation of attack techniques, and lack of attack technique coverage remain issues that make it difficult to compare across studies. Before the rise of graph neural networks, a large body of work in data mining and machine learning for intrusion detection confirmed that there were known challenges that were present in classical classifiers: feature engineering, class imbalance and trade-offs between detection latency and false-positive rates; and subsequent surveys of intrusion detection systems more broadly identified the aforementioned challenges as well as the mismatch between laboratory reported accuracy and the operating environment's tolerable false positive rate. APT detection surveys specifically targeted at long-dwell-time APTs identified the behavioral stages, tooling, and detection techniques of long-dwell-time APTs, confirming that there is no single data source that is enough on its own and multi-stage correlation is a structural necessity, not an optional feature. A general review of deep-learning based anomaly detection placed the graph-based intrusion analytics in the context of anomaly detection, noting that anomaly detection problems are becoming more common in production environments than scenarios with exhaustive labelling of attack data, and points out that increasingly, semi-supervised and self-supervised forms of the problem are being adopted [18]. Figure 2 overlays the methods and resources, discussed in Sections 2.1-2.3, on a common chronological axis.

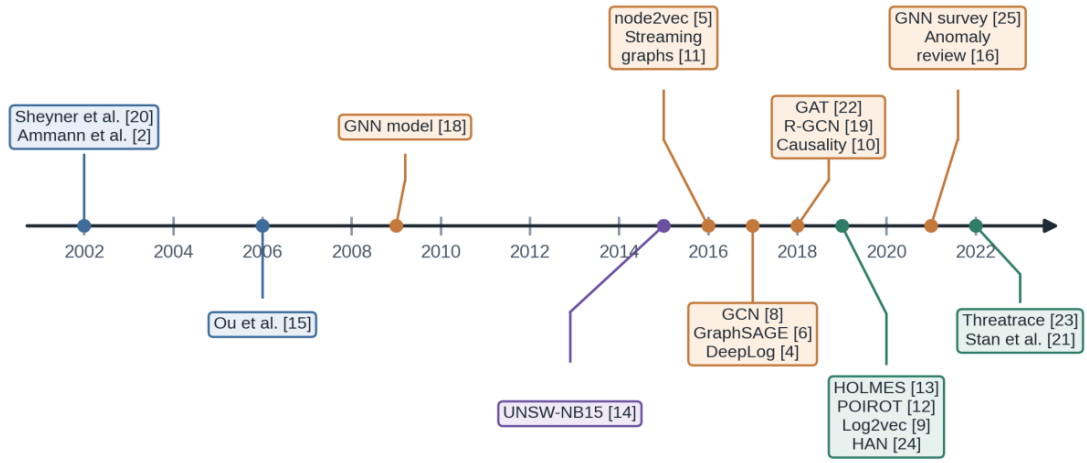


Figure 2. Timeline of Representative Graph-Based and Graph-Neural Approaches to Enterprise Threat Detection (2002–2022)

Markers indicate the publication year of representative studies discussed in Section 2; alternating placement above and below the axis is used only to avoid label overlap.

Table 4. Benchmark and Evaluation Resources Referenced in the Reviewed Literature

Resource	Type	Domain Coverage	Role in Reviewed Literature
UNSW-NB15 [22]	Labeled network-flow dataset	Modern synthetic + background traffic	Common network-intrusion benchmark
Dataset survey [23]	Meta-analysis of datasets	Cross-dataset coverage assessment	Guidance on limitations and class imbalance
ML/data-mining IDS survey [3]	Methodological survey	Classical and early learning-based IDS	Baseline of feature-engineering practice
IDS techniques survey [16]	Methodological survey	Signature, anomaly, hybrid IDS	Operational false-positive/latency benchmarks

Note. Resources are drawn from the reviewed literature rather than from independent benchmarking performed for this review.

3. METHODS

This review took place according to a focused methodological review, and not a broad bibliometric survey, process of which was structured but not registered. The corpus contained 26 peer-reviewed and archival sources from the formal attack-graph literature, from 2002 to 2022, as well as provenance-graph and log-based intrusion analytics, graph representation learning and graph neural network architectures, and network-intrusion benchmark datasets. Sources were selected by following the line of citations from three seed areas (attack-graph generation, provenance-based threat hunting, and graph neural network methodology), and included if they met two criteria: (a) the source was directly relevant to some aspect of graph-structured representation of security-relevant entities or events, and (b) the source provided sufficient methodological detail to enable the extraction of the architecture, mechanism, or dataset characteristics outlined in Section 2 [14].

For each source, the review identified the type of data representation (graph, sequence, or tabular), the learning paradigm (rule-based, statistical, shallow-embedding, or neural), and the task to be detected or ranked (either detection or ranking) and the benchmark resource used for evaluation (if applicable). These attributes were then aggregated into the four thematic strands outlined in Section 2 and cross-tabulated in Tables 1-4 to identify commonalities and the lack of commonalities between the thematic strands. The review proceeded qualitatively, as opposed to using the reported metrics from different studies and comparing them across different reported evaluation protocols, which would not be recommended for such a broad literature when the range of experimental designs and label definitions is large. This synthesis is then directly translated into the design rationale of the ThreatWeave-X conceptual framework, presented in Section 4 and in which every pipeline stage is related to a strand of literature that motivates them [10].

4. PROPOSED SYNTHESIS FRAMEWORK AND DISCUSSION

4.1 Rationale and Pipeline Overview

The literature reviewed in Section 2 indicates that none of the above techniques solve the multi-stage attack-path prediction problem as it is multi-stage; attack graphs represent reachability and are not observed; provenance-graph analytics represent observed behavior but are host-centric; and graph neural architectures provide powerful representation learning, but require a well-formed heterogeneous graph as input. The ThreatWeave-X framework is a conceptually combined pipeline of these strands, as depicted in Figure 3. Stage 1 normalizes and time-aligns signals from various sources in the end-to-end, network, identity and cloud space, mitigating the entity-resolution burden described throughout the literature on provenance-graphs. In Stage 2, a combined heterogeneous provenance graph is created where processes, files, network connections, user identities and hosts are modeled as typed nodes and relations as per schema conventions for relational and heterogeneous graph learning [11].

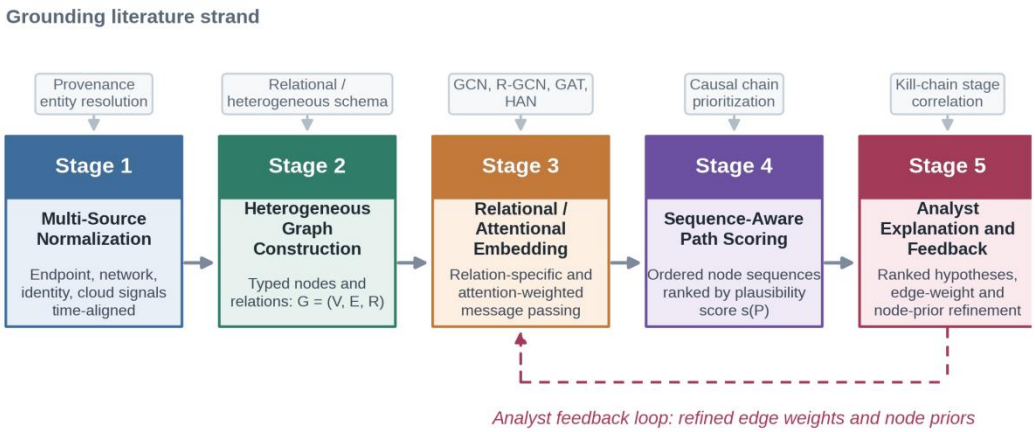


Figure 3. ThreatWeave-X Conceptual Processing Pipeline

The five stages trace directly to the literature strands reviewed in Section 2, culminating in an analyst-facing feedback loop.

This fusion process is illustrated schematically in fig. 4, as compared to the fragmentation of the raw signal sources to the unified typed-graph representation necessary for downstream learning. This is intentional: an attack graph is normally derived from a pre-built topology, in contrast to ThreatWeave-X, which starts with observed multi-source telemetry and builds an attack graph empirically [13].

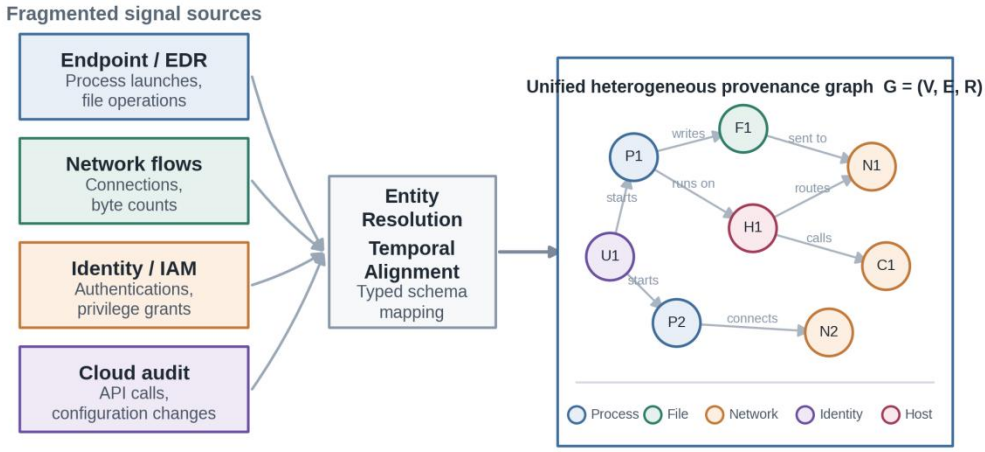


Figure 4. Fragmented Signals to Unified Provenance Graph

Entity resolution and temporal alignment precede typed node/edge schema mapping, yielding a heterogeneous graph G = (V, E, R).

4.2 Relational Embedding and Path Scoring

Stage 3 applies relational graph-neural embedding over the fused graph. Following the layer-wise propagation formalism established for convolutional graph learning [12], the hidden representation of node v at layer k + 1 is computed as

$$h_v^{(k+1)} = \sigma\left(W_{self}^{(k)} h_v^{(k)} + \sum_{r \in R} \sum_{u \in N_r(v)} (1 / c_{v,r}) W_r^{(k)} h_u^{(k)} \right) \quad (1)$$

where $N_r(v)$ denotes the neighbors of v under relation type r, $W_r(k)$ is a relation-specific weight matrix

consistent with the relational graph convolutional formulation, c_{vr} is a normalization constant, and σ is a nonlinear activation function. To capture varying relevance among neighbors of the same relation type, an attentional variant, consistent with the graph-attention formulation [22], replaces the fixed normalization with a learned coefficient:

$$\alpha_{v,u} = \text{softmax}_u (\text{LeakyReLU} (a^T [Wh_v \| Wh_u])) \quad (2)$$

so that the aggregated message becomes a weighted combination $\Sigma_u \alpha_{v,u} Wh_u$ rather than a uniformly or degree-normalized sum. Stacking K such layers yields a final node embedding $z_v = h_v(K)$, consistent with the multi-layer stacking pattern illustrated in Figure 5, which places representational capacity within reach of entities several relational hops from any single fragmented observation [7].

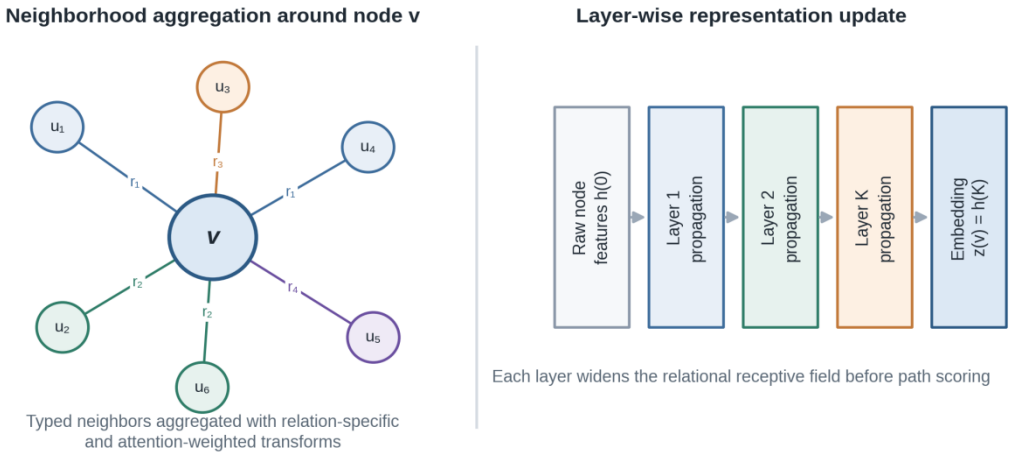


Figure 5. Neighborhood Aggregation and Layer-Wise Representation Update

Left: aggregation around a target node v across typed neighbors. Right: stacked propagation from raw features to the final embedding used for path scoring.

Stage 4 treats candidate multi-stage attack paths as ordered sequences of node embeddings P = (v1, v2, ..., vT) and assigns a plausibility score by combining pairwise transition compatibility with sequence-level attention, following the sequence-and-attention design pattern used for kill-chain-stage correlation [24]:

$$s(P) = (1 / (T - 1)) \Sigma_{t=1}^{T-1} \beta_t \cdot \cos (z_{v_t} , z_{v_(t+1)}) \quad (3)$$

where βt is a learned attention weight for transition t, and cos (·,·) represents cosine similarity of the successive node embeddings. The causal-chain prioritization principle demonstrated for provenance graphs with explicit backward and forward propagation of suspicion scores is formalized in embedding space in equation (3). Stage 5 also provides the analyst-facing hypotheses generated by the best paths, with their associated signals, and the analyst can adjust the weights on the edges and priors on the nodes, as is the case with purely automated APT-stage correlation, where the analyst needs to maintain the control of the number of alerts generated during the correlation process [23].

4.3 *Comparative Analysis and Cross-Cutting Challenges*

Table 1 helps to place the deterministic attack-graph lineage generated in Section 2.1 into a timeline, which exhibits a logical progression from exhaustive enumeration to more scalable, relational and infrastructure-aware formulations. Table 2 catalogs the provenance-graph and log based systems reviewed in Section 2.2, indicating that most previous systems

focus on one telemetry modality, while Table 3 shows the difference in representation-learning architectures synthesized in Section 2.3, between transductive and inductive, and between fixed or learned aggregation weights, with the latter paradigm being preferred because enterprise graphs grow continuously, so that induction and attention are preferable in Stage 3. Table 4 presents an overview of the benchmark and evaluation resources found in the literature surveyed, highlighting the current dependence on a few common datasets and potential problem of benchmark overfitting mentioned in the dataset survey literature [17]. Table 5 rounds out the synthesis by walking back the ThreatWeave-X pipeline stages to their grounding literature.

Table 5. Mapping of ThreatWeave-X Pipeline Stages to Literature Foundations

Pipeline Stage	Core Function	Grounding References
Stage 1	Multi-source normalization and entity resolution	[11], [14]
Stage 2	Heterogeneous provenance-graph construction	[20], [21]
Stage 3	Relational / attentional graph-neural embedding	[18], [19], [20]
Stage 4	Sequence-aware multi-stage path scoring	[10], [12]
Stage 5	Analyst-facing explanation and feedback loop	[1], [12]

Note. Grounding references indicate the strand of literature that motivates the design of each pipeline stage; they are not empirical validations of ThreatWeave-X itself.

A number of cross-cutting challenges arise from this synthesis. Firstly, the heterogeneity handling is only partially addressed: relational and heterogeneous architecture for attention have been suggested to address the naive graph convolution, but the majority of the reviewed systems involve only a single telemetry modality, and it remains unclear how embedding quality decreases as the modality count increases. Second, label scarcity remains as a structural constraint; in production environments, attack data is often only partially labeled and the abundance of semi-supervised and self-supervised formulations of classification are growing in popularity over fully supervised classification, as seen in the approach used in ThreatWeave-X that utilizes similarity in the embedding space instead of exhaustively labeled attack examples. Third, the lack of interpretability and the trust of the analysts are not solved yet: interpretability is identified as an open research direction in the comprehensive GNN surveys and hence is not satisfied only by a classification score, but has to be explicitly addressed as a path-explanation stage in ThreatWeave-X. Fourth, the different types of systems reviewed (host-only, network-only, and mixed telemetry) make it difficult to make direct quantitative comparisons and is the reason for this review's focus on architectural mechanisms, rather than the ability to aggregate incompatible reported metrics, as cautioned in earlier surveys of intrusion-detection datasets [20]. The four challenges above collectively outline the main direction for future empirical testing of graph-neural multi-stage attack-path prediction frameworks like ThreatWeave-X.

5. CONCLUSION

In this review, we collected 26 pieces of work across the fields of deterministic attack-graph generation, provenance-graph and log-based intrusion analytics, and graph neural network methodology to illustrate how the field of enterprise threat detection has been increasingly converging on graph representation learning as a common computational substrate. The synthesis revealed that attack-graph research was forced to move into the realm of relational and infrastructure-aware formulations, and that provenance-graph analytics shifted from sequence-only to explicitly relational and

causal representations of system behavior, while graph neural architectures transitioned from transductive (spectral convolution) to inductive, attentional, and relation-aware architectures for heterogeneous, continuously growing enterprise graphs. The paper builds on this synthesis and proposes ThreatWeave-X, a conceptual framework of five stages that integrates and combines fragmented endpoint, network, identity and cloud signals into a unified heterogeneous provenance graph, uses a relational and attentional graph-embedding and scores the candidate multi-stage attack paths using a sequence-aware attention mechanism, introducing an explicit analyst-feedback loop to ensure interpretability and continuous refinement. The framework is a literature-based architectural synthesis and not an empirically validated deployment; its main strength is in bringing together a disjointed collection of the methodology literature into a unified design rationale that needs to be further developed for future empirical testing in multi-modal enterprise data, systematic testing for scalability as modality and graph size increases, and further refinement of interpretable path-explanation mechanisms for security operations analysts.

ACKNOWLEDGEMENTS

The author(s) gratefully acknowledge the developers of the open-access studies synthesized in this review, whose methodological reporting made this synthesis possible. No dedicated funding was received for this review.

REFERENCES

[1] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019, doi: https://doi.org/10.1109/COMST.2019.2891891.

[2] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 217–224. doi: https://doi.org/10.1145/586110.586140.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015, doi: https://doi.org/10.1109/COMST.2015.2494502.

[4] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 1285–1298. doi: https://doi.org/10.1145/3133956.3134015.

[5] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864. doi: https://doi.org/10.1145/2939672.2939754.

[6] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proceedings 2002 IEEE Symposium on Security and Privacy*, 2002, pp. 273–284. doi: 10.1109/SECPRI.2002.1004377.

[7] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, in CCS '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 336–345. doi: 10.1145/1180405.1180446.

[8] O. Stan *et al.*, "Extending Attack Graphs to Represent Cyber-Attacks in Communication Protocols and Modern IT Networks," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 03, pp. 1936–1954, 2022, doi: 10.1109/TDSC.2020.3041999.

[9] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017. doi: https://doi.org/10.48550/arXiv.1706.02216.

[10] Y. Liu *et al.*, "Towards a timely causality analysis for enterprise security.," in *NDSS*, 2018, p. 141. doi: http://dx.doi.org/10.14722/ndss.2018.23254.

[11] E. Manzoor, S. M. Milajerdi, and L. Akoglu, "Fast memory-efficient anomaly detection in streaming heterogeneous graphs," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1035–1044. doi: https://doi.org/10.1145/2939672.2939783.

[12] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. N. Venkatakrishnan, "HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1137–1152. doi: 10.1109/SP.2019.00026.

[13] S. M. Milajerdi, B. Eshete, R. Gjomemo, and V. N. Venkatakrishnan, "Poirot: Aligning attack behavior with kernel audit records for cyber threat hunting," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 1795–1812. doi: https://doi.org/10.1145/3319535.3363217.

[14] F. Liu, Y. Wen, D. Zhang, X. Jiang, X. Xing, and D. Meng, "Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 1777–1794. doi: https://doi.org/10.1145/3319535.3363224.

[15] S. Wang *et al.*, "THREATRACE: Detecting and Tracing Host-Based Threats in Node Level Through Provenance Graph Learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3972–3987, 2022, doi: 10.1109/TIFS.2022.3208815.

[16] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, 2019, doi: https://doi.org/10.1186/s42400-019-0038-7.

[17] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009, doi: 10.1109/TNN.2008.2005605.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv Prepr. arXiv1609.02907*, 2017, doi: https://doi.org/10.48550/arXiv.1609.02907.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv:1710.10903, 2017. doi: https://doi.org/10.48550/arXiv.1710.10903.

[20] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," in *15th International Conference Semantic Web (ESWC 2018)*, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., Cham: Springer International Publishing, 2018, pp. 593–607. doi: https://doi.org/10.1007/978-3-319-93417-4_38.

[21] X. Wang *et al.*, "Heterogeneous Graph Attention Network," in *The World Wide Web Conference*, New York, NY, USA: Association for Computing Machinery, 2019, pp. 2022–2032. doi: 10.1145/3308558.3313562.

[22] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, Ieee, 2015, pp. 1–6. doi: https://doi.org/10.1109/MilCIS.2015.7348942.

[23] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, 2019, doi: https://doi.org/10.1016/j.cose.2019.06.005.

[24] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021, doi: 10.1145/3439950.