

Graph Neural Networks for Provenance-Based APT Detection under a Temporal Transfer Framework: A Review

Noor Hazlina Abdul Mutalib

Universiti Malaya

Aznul Qalid Md Sabri

aznulqalid@um.edu.my

Universiti Malaya

Ainuddin Wahid Abdul Wahab

Universiti Malaya

Erma Rahayu Mohd Faizal Abdullah

Universiti Malaya

Nouar AlDahoul

New York University Abu Dhabi

Research Article

Keywords: Graph neural networks, Provenance-based intrusion detection, Advanced persistent threats, Temporal transfer, Temporal data leakage

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11320684/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Graph Neural Networks for Provenance-Based APT Detection under a Temporal Transfer Framework: A Review

Noor Hazlina Abdul Mutalib^{1,4}, Aznul Qalid Md Sabri^{1,*}, Ainuddin Wahid Abdul Wahab², Erma Rahayu Mohd Faizal Abdullah¹, Nouar AlDahoul³

¹ Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

² Department of Computer System and Technology, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

³ Department of Computer Science, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates

⁴ Cybersecurity and Digital Trust, Malaysian Institute of Information Technology (MIIT), Universiti Kuala Lumpur, 50250 Kuala Lumpur, Malaysia

* Corresponding author: Aznul Qalid Md Sabri, Faculty of Computer Science and Information Technology, Universiti Malaya. E-mail: aznulqalid@um.edu.my

ORCID Noor Hazlina Abdul Mutalib: 0000-0002-5928-8593; Aznul Qalid Md Sabri: 0000-0002-4758-5400; Ainuddin Wahid Abdul Wahab: 0000-0003-1062-0329; Erma Rahayu Mohd Faizal Abdullah: 0000-0002-3026-9428; Nouar AlDahoul: 0000-0001-5522-0033

Abstract

Graph neural networks (GNNs) have become the dominant approach to detecting advanced persistent threats (APTs) in system provenance graphs, and studies that use them report near-perfect accuracy on public benchmarks. Recent re-evaluations show that these results weaken under consistent protocols and on newly generated attack data. This review examines how GNN-based provenance detectors handle time. We synthesise 29 detection studies and six evaluation or dataset studies published between 2020 and 2026. We define temporal transfer at two levels: the performance of a frozen detector with a locked threshold on later data, and the gain from adaptation over that frozen baseline. We propose a six-dimension taxonomy that covers graph construction, GNN type, supervision, detection unit, temporal handling and operating-point handling. Of the 29 detection studies, 19 use a GNN and 16 train on benign data only. Five describe an adaptation mechanism, but none evaluates it on data from a later period. Truncation of the graph at prediction time, the provenance of labels and the source of thresholds are rarely reported. We derive a 21-item reporting checklist and identify eight open problems, each paired with an experiment that would address it. We conclude that claims of detection on provenance graphs require prediction-time graphs, locked thresholds and evaluation on later periods.

Keywords Graph neural networks · Provenance-based intrusion detection · Advanced persistent threats · Temporal transfer · Temporal data leakage

1 Introduction

Advanced persistent threats (APTs) are long-lived, multi-stage intrusions in which an adversary gains a foothold, moves laterally and exfiltrates data while staying below the alerting thresholds of conventional defences (Alshamrani et al. 2019). Their individual steps often look benign in isolation, so detection depends on relating events that are separated by hours or days (Zhang et al. 2025). Provenance graphs meet this need: they record

processes, files and sockets as nodes and the system calls between them as directed, time-stamped edges, which preserves the causal history that backtracking analysis requires (King and Chen 2003). Systems built on this representation are known as provenance-based intrusion detection systems (PIDS; Zipperle et al. 2023).

Early PIDS encoded expert knowledge as rules over information flows, as in HOLMES (Milajerdi et al. 2019) and CONAN (Xiong et al. 2022). Learning-based detectors replaced rules with models of benign behaviour, first through graph sketches in UNICORN (Han et al. 2020) and then through graph neural networks (GNNs) in THREATTRACE (Wang et al. 2022). The current generation, represented by KAIROS (Cheng et al. 2024), reports precision and recall close to one on public benchmarks.

These scores have not yet translated into deployment. An industrial study found that practitioners remain concerned about alert volume, computational cost and the interpretability of provenance-based tools (Dong et al. 2023). A unified re-implementation of eight leading systems concluded that they were not viable for practical use and that a simple neural network outperformed them (Bilot et al. 2025). The gap between reported and achievable performance is a measurement problem as much as a modelling problem. The same gap between accuracy and operational value was described for learning-based network intrusion detection well before the current generation of models (Sommer and Paxson 2010).

This review argues that the missing variable is time. A detector is trained on one period and must raise alerts in a later one, during which benign workloads change and attackers alter their tooling. Security machine learning has documented the resulting decay for more than a decade, most rigorously in malware classification (Barbero et al. 2022).

Time also enters through the graph itself. Many studies summarise a node with features computed over the completed graph, such as final degree, lifetime or day-level counts. Such hand-crafted summaries threaten validity because they include events that occur after the moment at which an alert would have to be raised; this is the look-ahead bias described for temporal data in general (Lones 2024). The same quantities also leak label information when ground truth was propagated along the edges that the features summarise (Apicella et al. 2025). Applying a GNN to a graph truncated at prediction time is a direct way to test how much signal is available when the decision must be made.

Existing surveys do not address this intersection. Reviews of provenance-based detection predate most GNN detectors or organise the literature by pipeline stage (Zipperle et al. 2023). Reviews of GNNs for intrusion detection concentrate on network flows and treat host provenance briefly (Zhong et al. 2024). Reviews of concept drift in intrusion detection do not cover graph-structured host data (Shyaa et al. 2024). None of them asks how the primary studies handled chronological splits, threshold selection, label provenance or the availability of graph content at prediction time.

We organised the review around five research questions (RQs):

- **RQ1:** Which GNN and graph-learning designs have been used for provenance-based APT detection?
- **RQ2:** How do these studies construct graphs and partition data with respect to time?
- **RQ3:** How are operating points selected, locked and reported?
- **RQ4:** How were labels derived, and was the graph content used by the model available at prediction time?

- **RQ5:** Which adaptation mechanisms have been proposed?

1.1 Scope of the review

This review covers detectors that apply a GNN to host provenance or audit-log data in order to detect APTs. It includes studies published between 2020 and 2026 in peer-reviewed journals and major security or machine-learning conferences or workshops: 29 detection studies, of which 18 appeared in journals and 11 in conference proceedings, and six studies that evaluate such detectors or their datasets, listed in Appendix A. Each study is examined for what it reports about time, namely how the graph is constructed, how the data are partitioned, how the decision threshold is selected, how the labels were derived and whether the detector adapts after training.

1.2 Contributions

The paper makes the following contributions:

1. We propose a two-level definition of temporal transfer. Level 1 asks whether a frozen model retains its ranking quality and its locked operating point on later data. Level 2 asks whether adaptation improves on that frozen baseline. Separating ranking transfer from operating-point transfer explains how a detector can retain a high area under the curve while the number of alerts at its locked threshold exceeds the alert budget.
2. We present a six-dimension taxonomy covering graph construction, GNN type, supervision, detection unit, temporal handling and operating-point handling.
3. We provide a synthesis of 29 detection studies and six evaluation or dataset studies published between 2020 and 2026, with an evidence-gap map that shows where temporal claims remain untested.
4. We derive a reporting checklist for the prediction-time evaluation of provenance GNNs from the failure modes identified in the reviewed studies.

The remainder of the paper is organised as follows. Section 2 gives the background. Section 3 positions the review against earlier surveys. Section 4 describes the methodology. Section 5 presents the taxonomy. Section 6 reviews GNN detectors, and Section 7 reviews temporal transfer and adaptation. Section 8 examines datasets and evaluation practice. Section 9 discusses the findings and their limits. Section 10 sets out open challenges and future directions, and Section 11 concludes.

2 Background

2.1 Provenance graphs

A provenance graph is a directed, heterogeneous, time-stamped multigraph built from operating-system audit records. Nodes are subjects (processes) and objects (files, sockets, registry keys or pipes), and each edge is a typed event such as read, write, fork, execute or connect (Zipperle et al. 2023). The value of the representation is causal: an edge records that information could have flowed from one entity to another, so chains of edges link an entry point to its later effects (Pan et al. 2023).

Two properties of these graphs shape most subsequent design choices. The first is scale. A single host produces millions of events per day, and coarse-grained dependencies cause the number of plausibly related entities to

grow rapidly along a causal path, a problem known as dependency explosion (Inam et al. 2023). The second is that the graph is never complete. It grows edge by edge, and the graph that exists at time t contains only events whose timestamps do not exceed t :

$$G_t = (V_t, E_t), \quad E_t = \{(u, v, r, s) \in E : s \leq t\} \quad (1)$$

Here u and v are entities, r is the event type and s is the timestamp. We call a graph restricted in this way a prediction-time graph. A model that scores an entity at time t using any edge outside the prediction-time graph has been evaluated as a retrospective classifier, not as a detector.

2.2 Message passing on provenance graphs

GNNs compute a representation for each node by repeatedly aggregating messages from its neighbours (Gilmer et al. 2017). After k rounds, the representation of a node summarises its k -hop neighbourhood:

$$\mathbf{h}_v^{(k)} = \text{UPD}\left(\mathbf{h}_v^{(k-1)}, \text{AGG}\left\{\text{MSG}(\mathbf{h}_u^{(k-1)}, \mathbf{h}_v^{(k-1)}, \mathbf{e}_{uv}) : u \in \mathcal{N}(v)\right\}\right) \quad (2)$$

The main architectural families differ in how they aggregate. Graph convolutional networks use a fixed normalised sum (Kipf and Welling 2017), whereas GraphSAGE samples a bounded set of neighbours and is inductive, which allows it to score nodes that were absent during training (Hamilton et al. 2017). Graph attention networks learn a weight for each neighbour (Veličković et al. 2018). Relational variants keep separate parameters for each edge type (Schlichtkrull et al. 2018), and the heterogeneous graph transformer conditions attention on both node and edge types (Hu et al. 2020). These families are reviewed comprehensively elsewhere (Wu et al. 2021).

Temporal graph models add time to the aggregation. Discrete-time models process a sequence of snapshots, and continuous-time models consume a stream of time-stamped events (Skarding et al. 2021). Temporal graph attention encodes the time elapsed between events with a learnable functional time encoding (Xu et al. 2020), and temporal graph networks, proposed in a preprint by Rossi et al. (arXiv:2006.10637) and used as the encoder of KAIROS (Cheng et al. 2024), keep a memory vector per node that is updated as events arrive. Because attack labels are scarce, most provenance detectors train on benign data with a self-supervised objective and treat poorly modelled entities as anomalous (Liu et al. 2023). Masked graph autoencoders, which hide node features and learn to reconstruct them, are a frequent recent choice (Hou et al. 2022).

Message passing has two consequences for validity that are specific to provenance-based detection. A node score depends on its neighbours, so if the graph is not truncated the score depends on events from the future. In addition, when ground truth was produced by propagating an indicator along provenance edges, a GNN that propagates along the same edges can relearn the labelling rule instead of the attack behaviour. We return to both in Section 8.

2.3 The APT lifecycle

An APT campaign proceeds through reconnaissance, initial compromise, establishment of a foothold, privilege escalation, lateral movement and exfiltration, often over weeks (Alshamrani et al. 2019). Three consequences follow for detection. First, malicious entities are a very small fraction of all entities, so even a low false-positive

rate yields far more false alerts than true ones (Axelsson 2000). Second, the events of one campaign are strongly dependent, so the campaign, not the node, is the unit of statistical independence. Third, public datasets contain few campaigns, which limits what any evaluation can establish (Stojanović et al. 2020).

The operational cost of ignoring the first point is well documented. Analysts in security operations centres (SOCs) report that most of the alerts they receive are false or uninformative (Alahmadi et al. 2022), and the resulting alert fatigue is now treated as a research problem in itself (Tariq et al. 2025).

2.4 Two levels of temporal transfer

Concept drift is a change over time in the joint distribution of inputs and labels. Real drift alters the posterior and therefore the correct decision (Gama et al. 2014), whereas virtual drift alters only the input distribution (Bayram et al. 2022). In host provenance, three sources of drift are distinguishable. Benign drift, also called normality shift (Han et al. 2023), arises from software updates and new workloads and is the main threat to detectors trained only on benign data. Attack drift arises when adversaries change tools and techniques. Label drift arises when the procedure that assigns ground truth differs between periods.

We use the term temporal transfer for the performance of a detector on data collected after its training period, and we separate it into two levels (Fig. 1).

Level 1: frozen transfer.

The model is fitted on training days. Its threshold and any score calibrator are fixed on later calibration days and then locked. The locked detector is applied to still later test days. Two properties are measured separately. Ranking transfer asks whether threshold-free metrics, such as the area under the receiver operating characteristic curve (ROC-AUC) and average precision, are preserved. Operating-point transfer asks whether precision, recall and the number of false positives at the locked threshold are preserved. The distinction matters because a shift in the score distribution of benign entities can leave the ranking intact while multiplying the alerts raised at a fixed threshold.

Level 2: adaptive transfer.

The model, the threshold or both are updated with data that arrive after the lock, by periodic retraining, fine-tuning, replay or test-time adaptation. Level 2 results are interpretable only against the Level 1 baseline on the same test days.

This framing extends the constraints proposed for time-aware malware evaluation, which require that training data precede test data and that the test class ratio be realistic (Pendlebury et al. 2019). It adds the locked operating point as a first-class object and applies the constraints to graph construction as well as to data partitioning.

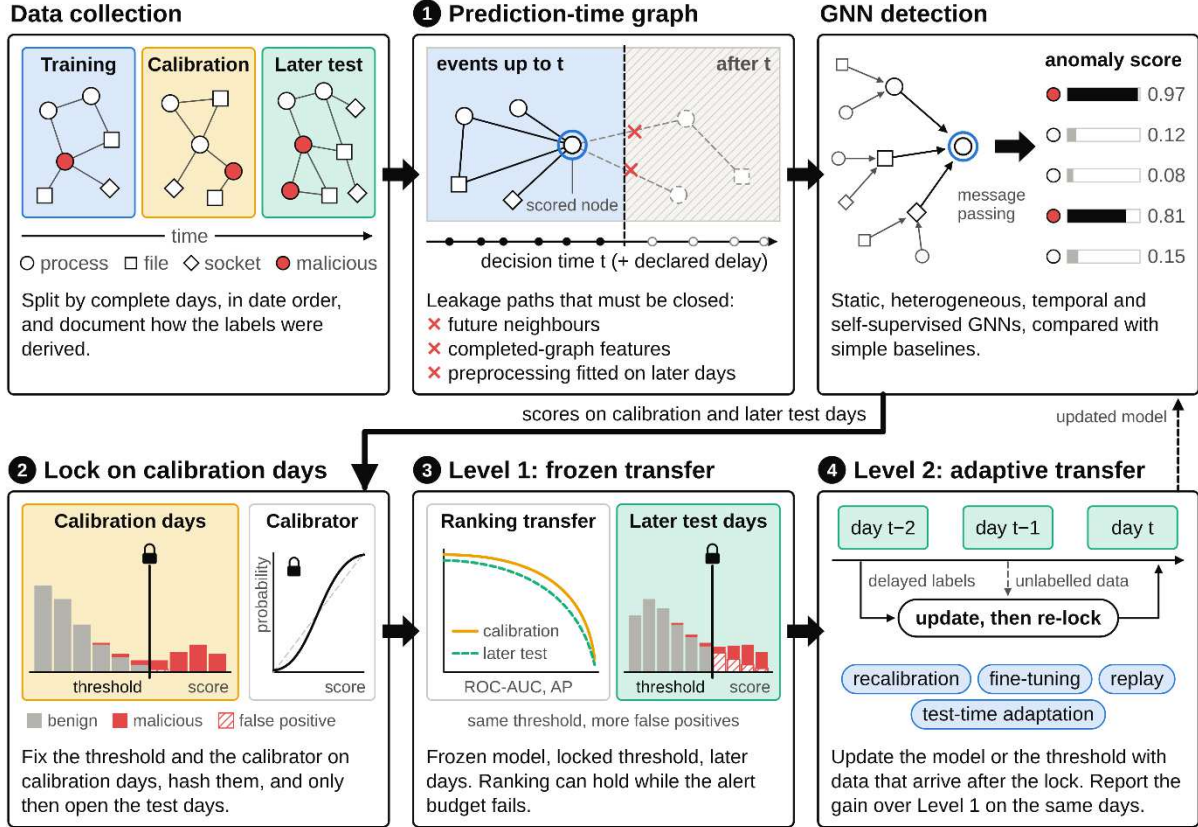


Fig. 1 Overview of the temporal-transfer framework for provenance-based intrusion detection. Data are split by complete days into training, calibration and later test periods. The numbered markers show four steps: (1) a graph truncated at the decision time t is scored by a GNN; (2) the threshold and calibrator are fixed on calibration days and locked; (3) Level 1 measures the frozen detector on later test days, separating ranking transfer from operating-point transfer; (4) Level 2 measures the gain from adaptation against that baseline. ROC-AUC, area under the receiver operating characteristic curve; AP, average precision

3 Related surveys

Four groups of surveys are adjacent to this review, and none is organised around the temporal transfer of graph-based provenance detectors (Table 1).

Provenance-based detection. An early review of threat detection and investigation with system-level provenance graphs structured the field into data collection, data management and analysis (Li et al. 2021). A taxonomy of PIDS covers data collection, graph summarisation, intrusion detection and benchmark datasets (Zipperle et al. 2023). Both appeared before most GNN detectors were published. Subsequent work systematised the auditing pipeline from capture and reduction to detection and investigation (Inam et al. 2023) and surveyed provenance across security and privacy applications (Pan et al. 2023). A recent pipeline-oriented survey, available as a preprint (Xu et al., arXiv:2504.07839), follows deep-learning detectors from log collection to attack investigation and lists concept drift among its challenges, but it does not analyse how drift is evaluated.

APT detection. One review organised APT detection techniques within a unified framework (Zhang et al. 2025), and another examined approaches, challenges and trends in APT detection systems (Buchta et al. 2024). A third reviewed explainable deep learning for APT detection across about 100 articles (Mutalib et al. 2024). These reviews are concerned with technique coverage and explanation. They do not examine the chronological

structure of the evaluations they summarise. That review, by the present authors, examined how deep-learning APT detectors explain their decisions. The present review asks a different question, the temporal validity of graph-based detectors on provenance data, and uses different inclusion criteria and a different analysis.

GNNs for intrusion and anomaly detection. GNNs for intrusion detection across network and host data were surveyed first in a general review (Bilot et al. 2023) and then with attention to methods, trends and challenges (Zhong et al. 2024). A further survey added an empirical comparison on a network dataset (Ismaeli Taba et al. 2026). A systematic review of 79 studies of dynamic temporal GNNs for cybersecurity anomaly detection identified scalability, limited real-world datasets and weak interpretability as the main barriers (Ares-Robledo et al. 2026). Broader surveys cover graph anomaly detection in general (Ma et al. 2023) and anomaly detection in dynamic graphs (Ekle and Eberle 2024). Another survey organises GNN-based anomaly detection by data characteristics and by assumptions about the distribution of anomalies (Yuan et al. 2025). In this group, time is treated as a model input rather than as a constraint on how the evaluation data are partitioned.

Concept drift and continual learning. One survey covers concept drift and feature dynamics in intrusion detection, with a focus on network data streams (Shyaa et al. 2024). Surveys of continual graph learning describe regularisation, replay and architectural methods for evolving graphs (Febrinanto et al. 2023), most recently with a benchmark comparison (Yuan et al. 2026). A further survey covers out-of-distribution generalisation on graphs (Li et al. 2025). These works are domain-general. They do not consider the conditions that make provenance data unusual: benign-only training, labels propagated along graph edges and decisions taken at a locked operating point.

Table 1 Related surveys and the gap addressed by this review

Survey	Scope	Provenance graphs	GNN detectors	Temporal transfer as an organising dimension
Li et al. (2021)	Provenance-based detection and investigation	Central	Predates most	No
Zipperle et al. (2023)	Taxonomy of PIDS	Central	Few	No
Inam et al. (2023)	Auditing pipeline, systematisation of knowledge	Central	Partly	No
Zhong et al. (2024)	GNNs for intrusion detection systems	Not explicit	Central	No
Mutalib et al. (2024)	Explainable deep learning for APT detection	No	Partly	No
Shyaa et al. (2024)	Concept drift in intrusion detection	No	No	Drift, non-graph data
Ares-Robledo et al. (2026)	Dynamic temporal GNNs for anomaly detection	Not specific	Central	Time as model input only
Ismaeli Taba et al. (2026)	GNNs for network intrusion detection	No	Central	No
This review	GNNs on provenance graphs for APT detection	Central	Central	Yes: splits, thresholds, labels, prediction-time graphs, adaptation

The gap addressed here is therefore specific. The literature offers taxonomies of provenance detectors and of GNN architectures, and a separate literature on drift. It does not offer an account of how GNN-based provenance detectors were evaluated with respect to time, or of what is known about their behaviour on data collected after training. General surveys of GNN learning paradigms (Waikhom and Patgiri 2023) and of heterogeneous GNNs (Bing et al. 2023) supply the architectural vocabulary used in Section 5, but they do not address this question either.

4 Review methodology

4.1 Information sources and search strategy

Six bibliographic databases were searched: Scopus, Web of Science, IEEE Xplore, the ACM Digital Library, SpringerLink and ScienceDirect. Backward and forward snowballing was applied to every included study. The search covered publications from 2020 to 2026. Earlier publications are cited as background and are not part of the corpus.

The core search string, adapted to the syntax of each database, was:

```
("provenance graph*" OR "audit log*" OR "system log*" OR "whole-system provenance") AND  
("graph neural network*" OR GNN OR "graph learning" OR "graph representation") AND  
("advanced persistent threat*" OR APT OR "intrusion detection" OR "threat detection")
```

A second pass added a temporal filter to collect direct evidence for the gap:

```
AND ("concept drift" OR temporal OR "time-aware" OR "continual learning" OR "transfer  
learning" OR "model aging")
```

4.2 Eligibility criteria

Table 2 lists the criteria. Network-only studies were excluded because flow graphs and authentication graphs lack the causal process, file and socket structure that defines provenance. A small number are nevertheless cited where they illustrate a temporal method.

Table 2 Eligibility criteria (IC, inclusion criterion; EC, exclusion criterion)

ID	Criterion
IC1	Published in a peer-reviewed journal or a major security or machine-learning conference or workshop; preprints dated 2025 or 2026 are admitted when they report an empirical evaluation, and are flagged as preprints
IC2	Written in English
IC3	Applies a GNN or another graph-learning model to provenance or audit-log data for threat detection, or evaluates such detectors or their datasets
EC1	Reports no empirical evaluation
EC2	Uses only non-graph models on non-graph features
EC3	Duplicate record, or superseded by a later version of the same work
EC4	Shorter than four pages
EC5	Uses only network-flow or authentication graphs without host provenance

4.3 Quality assessment

Each detection study was assessed against seven criteria that target the validity of temporal claims (Table 3). The criteria operationalise pitfalls catalogued for security machine learning, in particular data snooping, biased parameter selection and the base-rate fallacy (Arp et al. 2022). They also incorporate the temporal constraints proposed for malware evaluation (Pendlebury et al. 2019).

Table 3 Quality criteria (QC) for temporal validity

Criterion	Question asked of each study
QC1 Time-ordered partition	Do all training data precede all calibration and test data?
QC2 Threshold selection	Is the operating point fixed without reference to test data?
QC3 Label derivation	Is the ground-truth procedure documented, and is it independent of the model inputs?
QC4 Prediction-time availability	Is every edge and feature used for a decision available when the alert would be raised?
QC5 Realistic class ratio and unit	Is performance reported at the natural imbalance and for a stated detection unit?

Criterion	Question asked of each study
QC6 Variability	Are multiple seeds, or another estimate of uncertainty, reported?
QC7 Reproducibility	Are code, partitions and labels released?

4.4 Data extraction

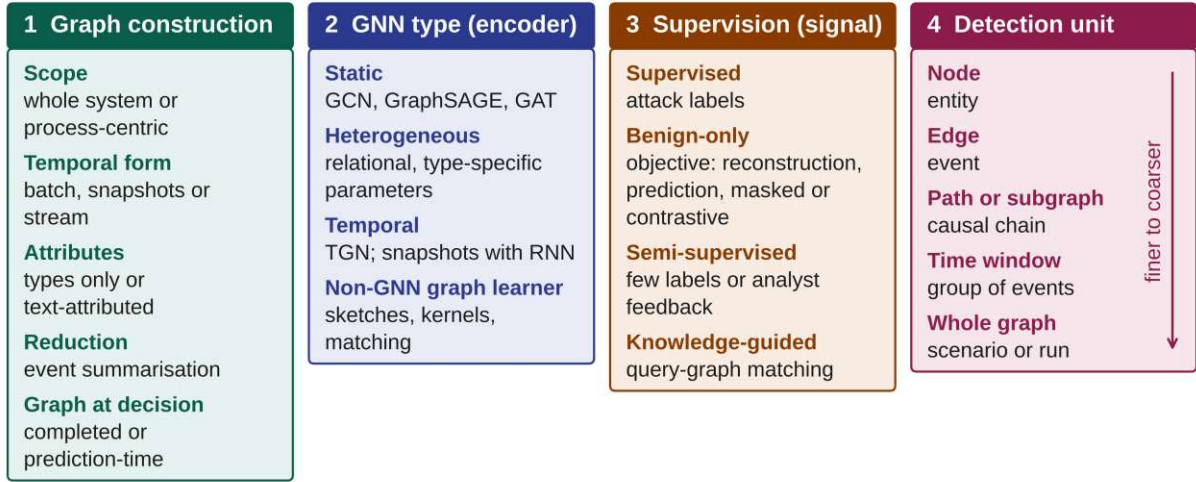
For each study, we extracted the graph construction method, node and edge types, model family, supervision regime, detection unit, datasets, partitioning rule, threshold rule, metrics, cost figures and any adaptation mechanism. We recorded what each publication reports. Where a publication is silent on a practice, the entry is "not reported"; we did not infer that the practice was absent.

The studies differ in datasets, ground-truth labels, detection units and metrics, so a quantitative meta-analysis was not attempted. The synthesis is narrative and is organised by the taxonomy of Section 5, with counts per category and an evidence-gap map in Section 9. Categories of data leakage follow an established classification into data-induced, preprocessing-related and split-related leakage (Apicella et al. 2025).

5 Taxonomy

The taxonomy has six dimensions (Fig. 2 and Table 4). The first four describe the design of a detector and refine categories used in earlier surveys. The last two describe how a study treats time and operating points, and they are where the evidence gap lies. The dimensions are independent: each answers a separate question, and a study takes one value on each. Dimension 2 records the encoder architecture, that is, how messages are aggregated, whereas dimension 3 records the training signal, that is, which labels the model sees and which objective it optimises. Masked and contrastive learning are therefore treated as objectives under dimension 3, not as architectures. KAIROS, for example, combines a temporal encoder (dimension 2) with benign-only training and an edge-reconstruction objective (dimension 3), and scores edges grouped into time windows (dimension 4).

a Detector design (dimensions 1–4)



b Temporal validity (dimensions 5–6, absent from earlier taxonomies)

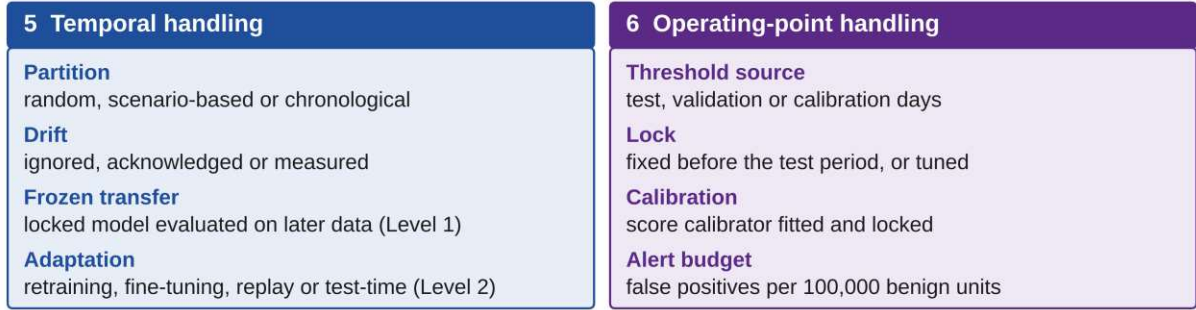


Fig. 2 Taxonomy of GNN-based provenance detection under temporal transfer. **a** Detector design, the four dimensions that refine earlier taxonomies; dimension 2 records the encoder architecture and dimension 3 the training signal, including the objective of benign-only models. **b** Temporal validity, the two dimensions absent from earlier taxonomies. GCN, graph convolutional network; GAT, graph attention network; TGN, temporal graph network; RNN, recurrent neural network

5.1 Graph construction

Studies consistently report four design choices. The first is scope: a whole-system graph, or a subgraph centred on a process. The second is temporal form. KAIROS consumes a stream of time-stamped events grouped into time windows (Cheng et al. 2024), whereas PROGRAPHER cuts the stream into a sequence of snapshots (Yang et al. 2023). The third is attribute encoding. Early GNN detectors used node and edge types only, whereas FLASH encodes process names, command lines and file paths with a word-embedding model before message passing (Ur Rehman et al. 2024). The fourth is reduction, which removes redundant events to control graph size.

A fifth choice is rarely stated: whether the graph given to the model for a decision at time t is the prediction-time graph defined in Section 2.1 or the completed graph for the whole collection period. We treat it as part of graph construction because it is fixed when the graph is built, before any model is trained.

5.2 GNN type

Static homogeneous models are the simplest family. THREATTRACE (Wang et al. 2022) and ProcSAGE (Xu et al. 2024) both build on GraphSAGE because neighbour sampling scales to millions of nodes. Relational and heterogeneous models keep type-specific parameters, as in APT-KGL (Chen et al. 2022). Temporal models encode the order and spacing of events, as in KAIROS (Cheng et al. 2024) and ORTHRUS (Jiang et al. 2025).

Non-GNN graph learners, such as the sketches of UNICORN (Han et al. 2020), are retained as comparators. Masked and contrastive models do not form a separate architecture: they pair one of these encoders with a self-supervised objective and are classified under supervision (Section 5.3).

5.3 Supervision

Four regimes appear in the corpus. Sixteen of the 29 detection studies train on benign data only and flag deviations. Seven are supervised with attack labels, three are semi-supervised, and three match query graphs derived from threat intelligence.

The benign-only objective takes several forms. THREATTRACE predicts the type of each node and treats misclassified nodes as anomalous (Wang et al. 2022). KAIROS reconstructs edges and scores them by reconstruction error (Cheng et al. 2024). Masked variants hide part of the graph and learn to reconstruct it (Ren and Geng 2025; Cui et al. 2026). Contrastive variants learn representations by comparing augmented views of benign graphs (Qiu et al. 2026a). The regime matters for temporal transfer because a benign-only model is, by construction, sensitive to any change in benign behaviour.

5.4 Detection unit

A detector may score nodes, edges, paths, subgraphs, time windows or whole graphs. Node-level detection is the most common unit in the corpus. PROVIDETECTOR scores causal paths (Wang et al. 2020), and OCR-APT scores subgraphs (Aly et al. 2025). UNICORN scores whole graphs (Han et al. 2020).

The unit determines what a metric means. A detector that flags one of 600 graphs correctly and a detector that flags the right process among millions solve different problems. Differences in detection granularity have been shown to inflate reported scores while making alerts less actionable (Bilot et al. 2026). ORTHRUS makes the same argument from the analyst's side and measures the number of nodes that must be inspected to attribute an attack (Jiang et al. 2025).

5.5 Temporal handling

This dimension has three parts. The partition may be random, scenario-based or chronological. Drift may be ignored or acknowledged as a limitation. Adaptation may be absent, or may take the form of periodic retraining, feedback-driven fine-tuning, continual learning or test-time adaptation. KAIROS reports chronological training, validation and test periods for the Defense Advanced Research Projects Agency (DARPA) Transparent Computing (TC) engagements (Cheng et al. 2024).

5.6 Operating-point handling

An anomaly score becomes an alert only when it crosses a threshold. This dimension records where the threshold comes from, whether it is locked before the test period, whether scores are calibrated, and whether results are reported at a stated alert budget. Thresholds derived from benign validation data are reported for KAIROS (Cheng et al. 2024). Slot adjusts thresholds dynamically with a multi-armed bandit (Qiao et al. 2025). Manually tuned thresholds have been identified as a form of data snooping in this literature (Bilot et al. 2026).

Table 4 Taxonomy dimensions, categories and representative studies

Dimension	Categories	Representative studies
1 Graph construction	Whole-system or process-centric; batch, snapshot or stream; typed or text-attributed; completed or prediction-time	Yang et al. (2023); Ur Rehman et al. (2024); Xu et al. (2024)
2 GNN type (encoder)	Static; relational or heterogeneous; temporal; non-GNN graph learner	Wang et al. (2022); Chen et al. (2022); Jiang et al. (2025)
3 Supervision (training signal)	Supervised; benign-only (reconstruction, prediction, masked or contrastive objective); semi-supervised; knowledge-guided	Alsaheel et al. (2021); Ren and Geng (2025); Zeng et al. (2022); Aly et al. (2024)
4 Detection unit	Node; edge; path; subgraph; time window; graph	Wang et al. (2020); Aly et al. (2025); Han et al. (2020)
5 Temporal handling	Partition; drift treatment; frozen transfer; adaptation	Amaru et al. (2025); Cui et al. (2026); Qiu et al. (2026a)
6 Operating-point handling	Threshold source; lock; calibration; alert budget	Cheng et al. (2024); Qiao et al. (2025); Bilot et al. (2026)

6 GNN detectors for provenance data

This section reviews the 29 detection studies by model family. Table 5 records their design, and Table 6 records their datasets, drawn mainly from StreamSpot, the DARPA TC engagements and the DARPA Operationally Transparent Cyber (OpTC) data, and what each publication reports about time. Nineteen studies use a GNN as the core learner. The remaining 10 use other learners over provenance graphs and serve as comparators. The model families used to organise this section and Tables 5 and 10 are a grouping for synthesis, not a further dimension. Most follow the encoder architecture (dimension 2), but the four self-supervised GNN studies are grouped by their masked, contrastive or one-class objective (dimension 3), because that objective, rather than their encoder, distinguishes them and makes them sensitive to benign drift (Section 6.5). Their supervision regime in Table 5 is therefore benign-only.

6.1 Precursors: sketches and kernels

The earliest learning-based detectors summarised a graph without a neural encoder. StreamSpot clusters sketches of streaming heterogeneous graphs and introduced the benchmark of the same name (Manzoor et al. 2016). UNICORN converts a growing provenance graph into fixed-size sketches of labelled neighbourhoods and models how those sketches evolve during long benign runs (Han et al. 2020). PROGRAPHER embeds successive snapshots and trains a sequence model to predict the next embedding (Yang et al. 2023). RT-APT recently returned to this family with a Weisfeiler–Lehman kernel and clustering, and reported lower runtime and memory use than earlier approaches (Weng et al. 2025).

These systems established two ideas that persist: training on benign data only, and treating the graph as a sequence in time. Their weakness is the detection unit. An alert on a whole graph tells an analyst that something is wrong but not where.

6.2 Static GNNs

Static GNNs moved detection to the node. PROV-GEM was an early supervised approach that classified provenance graphs from relation-aware embeddings (Kapoor et al. 2021). THREATTRACE trains GraphSAGE to predict the type of each node from its neighbourhood and treats nodes whose type is mispredicted as anomalous (Wang et al. 2022). FLASH adds a text encoder for node attributes and stores embeddings for reuse, which

reduces the cost of inference (Ur Rehman et al. 2024). ProcSAGE restricts attention to process nodes and reported reductions of 69 per cent in running time and 78 per cent in memory use relative to the StreamSpot baseline (Xu et al. 2024). Dinspector combines GraphSAGE with a dual-factor attention layer in a semi-supervised framework (Wang et al. 2026).

A second use of static GNNs is threat hunting, where a query graph built from threat intelligence is matched against the provenance graph. MEGR-APT learns attack representations for this matching (Aly et al. 2024). ProHunter uses a graph isomorphism network and reports full recall with a false-alarm rate below one per cent across 28 attack scenarios (Qiu et al. 2026b). Actminer performs the alignment without a neural model and updates it incrementally as new events arrive (Ma et al. 2025).

Two criticisms apply to the family. A static encoder ignores the order of events inside a neighbourhood, so two histories with the same edges and different orderings receive the same representation. In addition, node-type prediction is a proxy objective. It rewards the model for recognising what kind of entity a node is, which is related to, but not the same as, recognising that its behaviour is malicious.

6.3 Heterogeneous and knowledge-enriched models

Provenance graphs are heterogeneous by construction, and three studies exploit this directly. SHADEWATCHER maps audit records to a knowledge graph and applies a recommendation model that predicts how likely a process is to interact with an object; unlikely interactions are flagged (Zeng et al. 2022). APT-KGL combines a heterogeneous provenance graph with threat knowledge mined from public reports and learns meta-path-based embeddings for supervised process classification (Chen et al. 2022). A hierarchical heterogeneous attention network with parallel provenance storage reported an eight per cent gain in precision over earlier methods (Wu et al. 2025). Supervised heterogeneous models need attack labels. Their validity therefore depends on how those labels were produced.

6.4 Temporal graph networks

KAIROS applies a temporal graph network to whole-system provenance (Cheng et al. 2024). Its encoder maintains a state for each node, its decoder reconstructs each incoming edge, and edges with high reconstruction error are grouped into time-window queues that summarise an attack. ORTHRUS retains a spatio-temporal encoder but moves the detection unit to the node and adds dependency analysis, which reduced the number of nodes an analyst must inspect by several orders of magnitude (Jiang et al. 2025). A dynamic detector that couples a temporal graph network with a recurrent convolutional sequence model improved ROC-AUC over FLASH by 1.5 to 5.6 percentage points on three datasets (Wang et al. 2025a). EA-THGN adds heterogeneity and node-adaptive training to a temporal model and reported F1 scores up to 0.9998 on three DARPA TC hosts (Alotaibi et al. 2026).

Temporal encoders are also used on authentication graphs, where temporal link prediction scales to enterprise networks (King and Huang 2023). This work lies outside the corpus because it lacks host provenance, but it shows that snapshot encoders combined with recurrent models are computationally practical.

The temporal family encodes time as an input. It is tempting to assume that such models are therefore robust to change over time, but the assumption is untested. None of the four temporal GNN detectors in the corpus

describes an adaptation mechanism (Table 6), and node memories that accumulate state over long horizons are themselves a possible source of drift.

6.5 Masked and contrastive self-supervision

APT-MGL applies masked graph learning with a multi-head self-attention design and node-level outlier detection (Ren and Geng 2025). MGDA builds on a second-generation masked autoencoder and reported an average precision of 97.58 per cent with training up to three times faster than earlier methods (Cui et al. 2026). A contrastive alternative first identifies suspicious graphs and then localises anomalous nodes, with reported precision of 99.70 per cent on StreamSpot (Qiu et al. 2026a). OCR-APT applies one-class subgraph anomaly detection and uses a large language model to turn the flagged subgraphs into an attack narrative (Aly et al. 2025).

Reconstruction objectives have an inherent cost. A model that has learnt to reconstruct benign behaviour will reconstruct new benign behaviour poorly, so changes in workload tend to raise anomaly scores.

6.6 Hybrid, sequence-based and rule-based comparators

Several detectors operate on provenance graphs without message passing. ATLAS trains a sequence model on paths extracted from a causal graph to support attack investigation (Alsaheel et al. 2021). An intent-driven approach assigns tags to entities and classifies tag sequences, with a reported F1 score of 95.36 per cent (Yue et al. 2024). RAPID uses self-supervised sequence learning with embeddings that are updated iteratively, and reached 74 per cent precision at near-perfect recall with 30 per cent of the training data (Amaru et al. 2025). Slot embeds the graph with reinforcement learning under limited labels (Qiao et al. 2025).

Rule-based and view-based systems remain competitive on efficiency. E-Audit correlates threat-relevant events before classification (Patil et al. 2023), and BPGV matches behavioural graph views against learned benign patterns (Zipperle et al. 2026). Real-time rule-based systems such as CONAN (Xiong et al. 2022) and APTSHIELD (Zhu et al. 2023) have no training distribution at all, and therefore provide a stable reference point against which learned detectors should be justified.

6.7 Cross-cutting observations

Reported accuracy has saturated. Where the detection studies report a headline figure, it is typically above 95% precision, recall, F1 or average precision on StreamSpot, UNICORN or DARPA data (for example Cheng et al. 2024; Yue et al. 2024; Cui et al. 2026; Qiu et al. 2026a, b; Alotaibi et al. 2026); RAPID, at 74 per cent precision, is an exception (Amaru et al. 2025). These figures are as reported by each study and are not comparable across datasets or detection units, which is why Table 5 records design rather than results. Differences of one or two percentage points between architectures are difficult to interpret because run-to-run variability is rarely reported (Bilot et al. 2026). When eight systems were re-implemented under one protocol, a simple neural network outperformed them (Bilot et al. 2025).

Robustness is unresolved. Mimicry attacks that embed benign-looking activity around malicious processes evade several graph-based detectors (Goyal et al. 2023).

Time is a model input, not an evaluation axis. Temporal encoders, streaming graph construction and online processing are common. Evaluation of a locked detector on later data, as defined in Section 2.4, is not. Section 7 examines what is known.

Table 5 Design of the 29 detection studies, in order of publication year

Study	Family	Core learner	Supervision	Detection unit
UNICORN (Han et al. 2020)	Non-GNN	Graph sketches with evolutionary clustering	Benign-only	Graph
PROVDETECTOR (Wang et al. 2020)	Non-GNN	Rare-path embedding with outlier detection	Benign-only	Path
ATLAS (Alsaheel et al. 2021)	Non-GNN	Sequence model over a causal graph	Supervised	Entity
PROV-GEM (Kapoor et al. 2021)	Static GNN	Relation-aware graph embedding	Supervised	Graph
THREATTRACE (Wang et al. 2022)	Static GNN	GraphSAGE node-type prediction	Benign-only	Node
SHADEWATCHER (Zeng et al. 2022)	Heterogeneous GNN	Knowledge-graph embedding with GNN recommendation	Semi-supervised	Edge
APT-KGL (Chen et al. 2022)	Heterogeneous GNN	Meta-path embedding with threat knowledge	Supervised	Node
PROGRAPHER (Yang et al. 2023)	Non-GNN	Snapshot embedding with sequence prediction	Benign-only	Snapshot
E-Audit (Patil et al. 2023)	Non-GNN	Event correlation and classification	Supervised	Event
KAIROS (Cheng et al. 2024)	Temporal GNN	Temporal graph network encoder-decoder	Benign-only	Edge, time window
FLASH (Ur Rehman et al. 2024)	Static GNN	Text embeddings, GraphSAGE and a tree classifier	Benign-only	Node
ProcSAGE (Xu et al. 2024)	Static GNN	GraphSAGE with clustering	Benign-only	Node
MEGR-APT (Aly et al. 2024)	Static GNN	Graph matching on attack representations	Knowledge-guided	Subgraph
Yue et al. (2024)	Non-GNN	Tag-sequence semantic model	Supervised	Entity, event
ORTHRUS (Jiang et al. 2025)	Temporal GNN	Spatio-temporal encoder with dependency analysis	Benign-only	Node
Wang et al. (2025a)	Temporal GNN	Temporal graph network with a recurrent convolutional model	Benign-only	Graph
APT-MGL (Ren and Geng 2025)	Self-supervised GNN	Masked graph learning with self-attention	Benign-only	Node
RT-APT (Weng et al. 2025)	Non-GNN	Weisfeiler–Lehman kernel with clustering	Benign-only	Graph
Wu et al. (2025)	Heterogeneous GNN	Hierarchical heterogeneous attention network	Supervised	Graph
RAPID (Amaru et al. 2025)	Non-GNN	Self-supervised sequence model	Benign-only	Window, entity
Slot (Qiao et al. 2025)	Static GNN	Graph reinforcement-learning embedding	Semi-supervised	Node
OCR-APT (Aly et al. 2025)	Self-supervised GNN	One-class GNN with language-model reporting	Benign-only	Subgraph
Actminer (Ma et al. 2025)	Non-GNN	Query-graph alignment	Knowledge-guided	Subgraph
MGDA (Cui et al. 2026)	Self-supervised GNN	Masked graph autoencoder	Benign-only	Node, graph
Qiu et al. (2026a)	Self-supervised GNN	Multi-scale graph contrastive learning	Benign-only	Graph, then node
ProHunter (Qiu et al. 2026b)	Static GNN	Graph isomorphism network for graph matching	Knowledge-guided	Subgraph
Dinspector (Wang et al. 2026)	Static GNN	GraphSAGE with dual-factor attention	Semi-supervised	Node
EA-THGN (Alotaibi et al. 2026)	Temporal GNN	Temporal heterogeneous GNN	Supervised	Node
BPGV (Zipperle et al. 2026)	Non-GNN	Behavioural graph views with rule matching	Benign-only	Subgraph

Table 6 Datasets and temporal handling as reported by each detection study, in the order of Table 5

Study	Datasets	Drift or adaptation as reported
UNICORN	StreamSpot; DARPA TC E3; supply-chain simulations	Evolution modelled during training; model fixed at deployment

Study	Datasets	Drift or adaptation as reported
PROVDETECTOR	Enterprise hosts (private)	None identified
ATLAS	Ten simulated attacks	None identified
PROV-GEM	StreamSpot; UNICORN supply-chain data	None identified
THREATTRACE	StreamSpot; UNICORN; DARPA TC E3	None identified
SHADEWATCHER	DARPA TC E3; simulated attacks	Analyst feedback on false alarms used to update the model
APT-KGL	Two provenance datasets with simulated APT attacks	None identified
PROGRAPHER	StreamSpot; UNICORN; DARPA TC E3	None identified
E-Audit	Audit logs; enterprise test bed	None identified
KAIROS	StreamSpot; UNICORN; DARPA TC E3 and E5; OpTC	Chronological partition; thresholds from validation data
FLASH	DARPA TC E3; OpTC; StreamSpot; UNICORN	None identified
ProcSAGE	StreamSpot	Drift acknowledged as a limitation
MEGR-APT	DARPA TC E3; OpTC	None identified
Yue et al. (2024)	Ten public APT datasets	None identified
ORTHRUS	DARPA TC E3 and E5	None identified
Wang et al. (2025a)	StreamSpot; CamFlow-APT; ShellShock	None identified
APT-MGL	DARPA TC E3; StreamSpot	Adaptation through benign feedback
RT-APT	Laboratory data; StreamSpot; UNICORN	Streaming input; none identified
Wu et al. (2025)	Five datasets	None identified
RAPID	DARPA TC E3 hosts; OpTC; one further public set	Embeddings updated iteratively without retraining
Slot	Three public datasets, including DARPA TC E3	Thresholds adjusted by reinforcement learning
OCR-APT	DARPA TC E3; OpTC; NODLINK data	None identified
Actminer	DARPA TC engagements	Incremental alignment of new events; none identified for drift
MGDA	StreamSpot; UNICORN; DARPA TC E3	Feedback retraining proposed as future work
Qiu et al. (2026a)	StreamSpot; UNICORN; DARPA TC	Periodic low-overhead retraining
ProHunter	DARPA TC E3 and E5; OpTC	None identified
Dinspector	Three public datasets	None identified
EA-THGN	Three DARPA TC E3 hosts	None identified
BPGV	DARPA OpTC	Offline analysis

"None identified" means that no drift or adaptation mechanism was found in the published description. It does not establish that the authors did not consider drift. DARPA, Defense Advanced Research Projects Agency; TC, Transparent Computing; E3 and E5, Engagements 3 and 5; OpTC, Operationally Transparent Cyber

7 Temporal transfer and adaptation

7.1 Evidence on the transfer of provenance detectors

Direct evidence comes from a recent study of the benchmarks themselves. It found that the DARPA TC data rely on scripted and repetitive benign workloads, and that the OpTC data contain undocumented distribution shifts between the training and test periods (Cai et al. 2026). When eight detectors were run on newly generated attack chains, several reached near-perfect recall with near-zero precision at node level.

These findings agree with the wider record. Benign-only anomaly detectors in three security applications degraded under normality shift in long-term operational data (Han et al. 2023). Malware classifiers evaluated with time-ordered splits perform substantially worse than when they are evaluated with random splits (Pendlebury et al. 2019). What is missing for provenance GNNs is a study that reports ranking transfer and operating-point transfer separately across successive periods.

7.2 Level 1: frozen-model transfer

A Level 1 study needs four elements: a chronological partition into training, calibration and test days; a lock of the model, threshold and calibrator before the test days are opened; separate reporting of threshold-free and fixed-threshold metrics; and an estimate of uncertainty over days or campaigns.

The separation of metrics is the central point. ROC-AUC and average precision depend only on the ordering of scores, so they are unchanged by any monotonic transformation of the scores. Precision, recall and the false-positive count at a fixed threshold are not. For rare positives, precision-recall analysis is more informative than ROC analysis (Saito and Rehmsmeier 2015). The Matthews correlation coefficient gives a single summary of the confusion matrix at the locked threshold that remains meaningful under imbalance (Chicco and Jurman 2020).

Calibration deserves the same treatment as the threshold. Post-hoc calibrators are fitted on held-out data and assume that the deployment distribution matches it (Silva Filho et al. 2023). Under drift that assumption fails, so a calibrator must be locked with the threshold and its error reported on later data.

In the corpus, few studies derive thresholds from validation data within a chronological partition, and KAIROS is the most explicit example (Cheng et al. 2024). It reports results for the test period as a whole and does not report how the false-positive count at the locked threshold changes across successive later periods, so the rate of decay is unknown.

7.3 Level 2 in the corpus: adaptation mechanisms

Five of the 29 detection studies describe a mechanism that changes the detector after initial training. They fall into four groups.

Analyst feedback. SHADEWATCHER retrain on interactions that analysts mark as false alarms (Zeng et al. 2022). APT-MGL feeds confirmed benign samples back into the model (Ren and Geng 2025). Feedback is cheap to implement, but it is available only for entities that triggered alerts, and it depends on analyst attention, which is the very resource the detector is meant to conserve.

Periodic retraining. One contrastive detector proposes periodic, low-overhead retraining on recent benign data (Qiu et al. 2026a). The approach assumes that recent data are benign. An adversary already present in the environment can violate that assumption, and their activity may then be absorbed into the model of normal behaviour.

Incremental representation updates. RAPID updates embeddings iteratively during inference so that the model follows changing behaviour without full retraining (Amaru et al. 2025).

Adaptive thresholds. Slot adjusts decision thresholds with a multi-armed bandit instead of fixing them (Qiao et al. 2025).

None of these mechanisms has been evaluated across a dated shift (that is, on data from an explicitly later period), and no detector in the corpus has been evaluated at Level 2 against a locked Level 1 baseline. Table 7 sets these mechanisms beside the strategies available from adjacent fields, which Section 7.4 reviews.

Table 7 Adaptation strategies, their requirements and the evidence available for provenance graphs

Strategy	What is updated?	Labels required	Main risk	Evidence on provenance graphs
Frozen model (Level 1 baseline)	Nothing	None	Decay of precision or recall	None in the corpus
Periodic retraining on recent benign data	All parameters	None, but data are assumed benign	Poisoning by a resident adversary; forgetting	Described, not evaluated across a shift (Qiu et al. 2026a)
Analyst-feedback retraining	All or some parameters	Confirmed false alarms	Selection bias; analyst cost	Described (Zeng et al. 2022)
Fine-tuning with frozen layers	Upper layers and a new head	A sample of post-shift data	Forgetting if too many layers are released	None in the corpus
Replay or regularisation	Parameters, with a memory buffer or penalty	Depends on objective	Memory cost; stale exemplars	None in the corpus
Test-time adaptation	Normalisation statistics or a few parameters	None	Error accumulation; adversarially induced drift	None in the corpus
Threshold re-estimation or recalibration	Threshold or calibrator only	Recent benign scores	Masks decay in ranking	Adaptive thresholds described (Qiao et al. 2025)
Drift detection with rejection	Nothing; uncertain inputs are deferred	None	Workload moves to analysts	None in the corpus

7.4 Methods available from adjacent fields

Four bodies of work offer methods that have not yet been tested on provenance graphs.

Drift detection. Conformal evaluation rejects predictions on inputs that fall outside the calibrated region, which turns drift into a controlled abstention rate (Barbero et al. 2022). A contrastive drift detector with a nearest-class-mean classifier has been shown to remain stable when an attacker deliberately induces drift (Kuppa and Le-Khac 2022). For benign-only detectors, the OWAD framework detects a shift in the distribution of calibrated model outputs, explains it and adapts with a limited labelling budget (Han et al. 2023).

Continual learning. Methods are commonly grouped into regularisation, replay and architectural approaches (Wang et al. 2024). Elastic weight consolidation is the best-known regulariser: it penalises changes to parameters that were important for earlier data (Kirkpatrick et al. 2017). In security, continual learning with hierarchical contrastive representations and active sample selection reduced the labelling needed to keep an Android malware classifier current (Chen et al. 2023). INSOMNIA combined incremental learning with active learning and pseudo-labels for network intrusion detection (Andresini et al. 2021). A multi-agent framework used continual anomaly detectors with federated knowledge distillation to adapt local models to changing traffic (Soltani et al. 2024).

Continual graph learning. On graphs, new nodes arrive connected to old ones, so the tasks are not independent and forgetting interacts with topology (Febrinanto et al. 2023). A recent benchmark-based survey found that early work favoured regularisation, that later work favours replay and architectural strategies, and that regularisation struggles when new classes keep arriving (Yuan et al. 2026). Surveys of dynamic GNNs list adaptive and memory-enhanced models among their open problems (Zheng et al. 2025).

Test-time adaptation and distribution shift on graphs. Tent adapts a deployed model without labels by minimising the entropy of its predictions (Wang et al. 2021). A survey of online test-time adaptation re-benchmarks the main variants and finds that their effectiveness depends on conditions such as the backbone and

the batch size (Wang et al. 2025b). For graphs, out-of-distribution generalisation is surveyed as a problem of data, model and learning strategy (Li et al. 2025).

None of these methods transfers directly, for four reasons. Labels in this domain are scarce and arrive late, often only after an incident investigation. Most provenance detectors are trained on benign data alone, so objectives that rely on class predictions, such as entropy minimisation, have no direct counterpart. Nodes and edges in one graph are not independent samples, which complicates replay and rejection. Finally, the adversary can influence the data that an adaptive detector learns from, so every Level 2 mechanism has to be evaluated for poisoning as well as for accuracy. Drift detection research has begun to test robustness to attacker-induced drift (Kuppa and Le-Khac 2022); provenance detection has not.

8 Datasets and evaluation

8.1 Datasets

A small number of datasets account for almost all of the evidence in Table 8. At least 17 of the 29 detection studies evaluate their detectors on DARPA TC data, and 12 on StreamSpot (Table 6). The concentration matters because the properties of these datasets bound what any study can claim about time.

StreamSpot contains 600 graphs from six scenarios, five benign and one attack, with labels at graph level (Manzoor et al. 2016). The graphs have no shared timeline, so the dataset cannot support a chronological partition. The DARPA TC engagements provide time-stamped audit data from several operating systems during red-team exercises (DARPA 2020). Their benign activity is scripted and repetitive, and their attack scenarios are now outdated (Cai et al. 2026). The OpTC data are larger and more operationally realistic, and their usefulness for detection research has been analysed separately (Anjum et al. 2021).

Three limits recur across datasets. Temporal span is short, from 6 to 10 days for the DARPA engagements and OpTC (Table 8), so a chronological partition leaves few independent days in each part. Campaigns are few, so a test period may contain one or two scenarios. Ground truth is coarse or inconsistent, and labelling granularity differs between datasets and between the studies that use them (Cai et al. 2026). An earlier review of APT datasets reached the same conclusion about scarcity before most of the GNN detectors were published (Stojanović et al. 2020).

A newer collection addresses some of these limitations. The ProvAttack datasets combine automated attack emulation with benign traces from volunteers and software agents, and they label individual entities with timestamps and technique identifiers (Cai et al. 2026). They do not yet span enough time to study drift across months.

Table 8 Public datasets used by the corpus, their time spans and their limits for temporal-transfer studies

Dataset	Source	Content	Labels	Time span and limit for temporal transfer
StreamSpot	Manzoor et al. (2016)	600 Linux graphs; five benign scenarios and one attack scenario	Graph level	No shared timeline (independent runs)
DARPA TC Engagement 3	DARPA (2020)	Red-team engagement on several operating systems	Ground-truth report; entity labels reconstructed by researchers	8 days (6–13 April 2018); scripted benign activity

Dataset	Source	Content	Labels	Time span and limit for temporal transfer
DARPA TC Engagement 5	DARPA (2020)	Later engagement with similar hosts	As above	10 days (8–17 May 2019); supports one cross-engagement shift
DARPA OpTC	Anjum et al. (2021)	Large Windows enterprise telemetry with a red-team period	Red-team ground-truth report	6 days (19–25 September 2019), 3 with red-team activity; undocumented shifts between training and test periods
UNICORN supply-chain	Han et al. (2020)	Simulated supply-chain attacks captured with whole-system provenance	Graph level	No shared timeline (independent runs)
ATLAS	Alsaheel et al. (2021)	Ten simulated attacks	Entity level	Separate per-scenario traces; labelling granularity varies between scenarios
ProvAttack	Cai et al. (2026)	Emulated multi-stage attacks; human and agent benign traces	Entity level with timestamps and technique labels	Single attack chains

8.2 Prediction-time graph construction

A detector raises an alert at a decision time t , and everything it uses must exist at t (Fig. 3). We identify three paths by which later information reaches the decision.

1. Future neighbours. If message passing runs over the completed graph, the k -hop neighbourhood of a node includes edges created after t . A process whose malicious child is spawned an hour later is then scored with knowledge of that child.
2. Completed-graph features. Final degree, lifetime and day-level counts summarise the whole collection period. These are the hand-made summaries discussed in Section 1.
3. Global preprocessing. Vocabularies, text embeddings and normalisation constants are often fitted on all days, including test days. This is preprocessing-related leakage in an established classification of data leakage (Apicella et al. 2025).

A fourth, related problem is that identifiers can persist across partitions. Process and file identifiers, host names and date fields allow a model to memorise entities, which is one form of the spurious correlation pitfall in security machine learning (Arp et al. 2022).

Few studies state how the graph was truncated. Streaming designs process events in time order and score them within windows, so the end of the window is a natural decision time (Cheng et al. 2024). For node-level detectors that operate on a graph of a whole day or scenario, the decision time is undefined, and the reported scores are retrospective.

The remedy is to construct the graph at t , or at t plus a declared delay, and to report performance as a function of that delay. Doing so makes timeliness measurable. The time-weighted detection accuracy proposed with the ProvAttack datasets, which gives more credit to early detections, follows the same reasoning (Cai et al. 2026). A leakage audit of this kind should precede model training because look-ahead bias cannot be removed after the fact (Lones 2024).

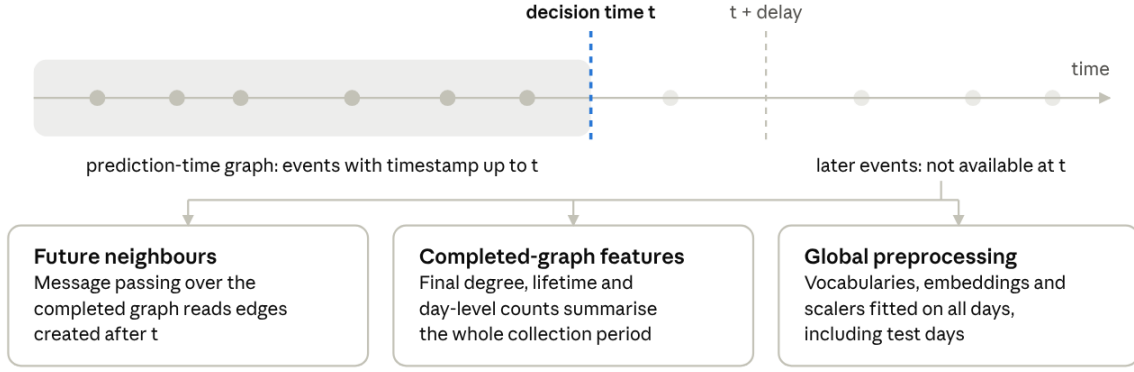


Fig. 3 Prediction-time graph construction. Events to the left of the decision time t form the prediction-time graph. The three boxes are the paths by which later events leak into a decision

8.3 Label provenance

Ground truth for provenance data is typically produced in one of three ways. Red-team reports are mapped to entities by the researchers who use the data, and different groups have produced different mappings from the same report. Indicators are matched to a few entities and then propagated to related entities along provenance edges. Labels are assigned to a whole graph or time window. Incompatible ground-truth labelling between studies is one of the documented obstacles to reproducibility in this field (Bilot et al. 2026).

Propagated labels create a specific hazard for GNNs. If a node is labelled malicious because it lies within a few hops of an indicator-matched node, then the label is a function of graph structure. A GNN aggregates over the same structure and can recover the labelling rule. Its test score then measures agreement with the rule, not detection of the attack. This is data-induced leakage, because the information enters during dataset construction (Apicella et al. 2025). We found no detection study in the corpus that reports performance stratified by graph distance from the indicator-matched nodes, which is the analysis that would expose it.

Two practices reduce the risk. Conservative labelling marks only entities that participate directly in the attack and avoids transitive propagation (Cai et al. 2026). Consistent treatment of contaminated or ambiguous nodes across all compared systems prevents the labelling choice from favouring one of them. Benign-only training removes labels from the training objective, but it does not remove them from the evaluation, so the hazard persists whenever propagated labels define the test set.

8.4 Thresholds, metrics and uncertainty

Thresholds. Selecting a threshold with reference to test data is the pitfall of biased parameter selection (Arp et al. 2022). Manually tuned thresholds are still found in this literature, as Section 5.6 noted. Validation-derived thresholds are a necessary improvement, but they are not sufficient for claims about deployment, because a threshold that is valid on the validation days may not hold its alert budget on later days.

Metrics. We recommend reporting two groups of metrics separately. The first group is threshold-free: ROC-AUC and average precision. The second is measured at the locked threshold: precision, recall, the Matthews correlation coefficient and a workload measure, such as the number of false positives per 100,000 benign nodes. The workload measure matters because the cost of a detector to a SOC is paid in analyst time (Alahmadi et al. 2022). Attribution quality, measured as the number of nodes an analyst must inspect, complements these metrics

(Jiang et al. 2025). Coverage of attack techniques at a minimum quality level reveals blind spots that aggregate recall hides (Cai et al. 2026).

Uncertainty. Neural detectors vary between random seeds, and single-run comparisons are unreliable. Seeds capture only one source of variation. With few collection days and one or two campaigns in the test period, the dominant uncertainty is between days and between campaigns, and it should be estimated by resampling those units. Repeated evaluation on the same test days is a further risk: each look at the test set, followed by a design change, moves the model towards that set (Lones 2024).

8.5 A reporting checklist

Table 9 translates the failure modes of Sections 8.2 to 8.4 into a checklist of 21 items. It is intended for authors and reviewers of studies that claim detection, as opposed to retrospective classification, on provenance graphs. The checklist extends general guidance on machine learning in security (Arp et al. 2022) with items that are specific to graph construction, propagated labels and locked operating points. Groups G, P, L and O correspond to quality criteria QC1 to QC5 in Table 3, and group R corresponds to QC6 and QC7. Group T has no counterpart among the quality criteria; it covers the reporting of Level 1 and Level 2 results defined in Section 2.4.

Table 9 Reporting checklist for prediction-time evaluation of provenance GNNs

Item	Requirement
G1	State the decision time for each scored unit and build the graph from events up to that time
G2	If a delay is allowed, declare it and report results for each delay
G3	Fit vocabularies, embeddings and scalars on training days only
G4	Remove or anonymise identifiers that persist across partitions, such as host, date and entity identifiers
P1	Use a chronological partition into training, calibration and test days, and give the dates
P2	Report the number of days and of campaigns in each part
P3	Keep a final block of test days unopened until the model and threshold are locked and hashed
L1	Document how ground truth was derived, including any propagation rule and its hop limit
L2	Report performance by graph distance from indicator-matched entities
L3	Apply the same labels and exclusions to every compared system
O1	Select the threshold and calibrator on calibration days only, and lock them
O2	Report threshold-free and fixed-threshold metrics separately
O3	Report false positives per 100,000 benign units at the locked threshold, against the nominal alert budget
O4	Report calibration error on later days
T1	Report Level 1 results for each later period, not only pooled
T2	Report any Level 2 method against the Level 1 baseline on the same days
T3	State which data and labels the adaptation consumed, and test it against poisoned updates
R1	Report variation over seeds and over days or campaigns
R2	Compare with simple baselines, such as an allowlist and a tree ensemble on prediction-time features
R3	Report training time, memory and inference latency
R4	Release code, partitions, labels and locked thresholds

9 Discussion

9.1 Answers to the research questions

RQ1: Graph-learning designs.

19 of the 29 detection studies use a GNN: eight static, four self-supervised, four temporal and three heterogeneous. Ten use other learners over provenance graphs. Sixteen studies train on benign data only. The field has moved from graph-level sketches in 2020 to node-level temporal and masked models between 2024 and 2026. By year of publication, the corpus contains two detection studies in each of 2020, 2021 and 2023, three in 2022, five in 2024, nine in 2025 and six in 2026 up to the search date. These counts reflect the choice of venue as well as research activity, because most early work appeared at security conferences, and the figure for 2026 covers only part of the year.

RQ2: Graph construction and partitions.

Time is widely used as a model input and rarely used as an evaluation constraint. Chronological training, validation and test periods are seldom reported explicitly. Whether the graph given to the model was truncated at the decision time is seldom stated. Twelve studies evaluate on StreamSpot, which has no shared timeline and therefore cannot support a chronological partition.

RQ3: Operating points.

Thresholds are derived from validation data in a few studies and adapted online in another; elsewhere their source is seldom stated, and manual tuning has been documented in this literature (Bilot et al. 2026). We found no study that reports the false-positive count at a locked threshold across successive later periods, and none that reports score calibration.

RQ4: Labels and availability.

Labels are reconstructed from red-team reports or propagated along edges, and the procedure differs between studies that use the same data. We found no analysis of performance by graph distance from indicator-matched entities. Leakage through global preprocessing is not discussed in the detection studies.

RQ5: Adaptation mechanisms.

Five studies describe an adaptation mechanism, and two acknowledge drift as a limitation or as future work. For the remaining 22, no mechanism was identified. No study evaluates adaptation across a dated shift, and no provenance detector has been evaluated at Level 2 against a locked Level 1 baseline.

9.2 Evidence-gap map

Table 10 cross-tabulates model family against the treatment of drift. Two cells stand out. The temporal GNN family, which might be expected to engage most with change over time, has no study that describes an adaptation mechanism. The self-supervised family engages most, with three of four studies either describing a mechanism or naming drift as a limitation. This is consistent with the sensitivity of reconstruction objectives to benign change noted in Section 6.5. No study in any family has evaluated its mechanism on data from an explicitly later period (Section 7.3), so whether any of these mechanisms works on later data has not been tested.

Table 10 Evidence-gap map: number of detection studies by model family and treatment of drift

Model family	Studies	Adaptation mechanism described	Drift acknowledged only	None identified
Non-GNN graph learners	10	1	0	9
Static GNN	8	1	1	6
Heterogeneous GNN	3	1	0	2
Temporal GNN	4	0	0	4
Self-supervised GNN	4	2	1	1
Total	29	5	2	22

9.3 Implications for practitioners

Four implications follow for teams that evaluate or deploy these detectors.

First, benchmark scores should be read as retrospective upper bounds. A score obtained on a completed graph, with a threshold informed by the test period, says little about the alert stream a deployed detector would produce.

Second, a Level 1 test on local data is inexpensive and informative. Train on earlier weeks, set and lock the threshold on the following days, and count alerts per analyst-day on later weeks. If the count exceeds what the team can triage, the ranking quality of the model is of little practical value. Alert volume, computational cost and interpretability are the concerns that practitioners themselves raise about provenance-based tools (Dong et al. 2023).

Third, every learned detector should be compared with a simple baseline on the same later data. Simple models have matched complex ones under a controlled protocol (Bilot et al. 2025).

Fourth, adaptation needs safeguards. The distribution of benign scores is an early signal of normality shift and can be monitored without labels. Updates driven by recent data or analyst feedback should be tested against a frozen reference model, so that a gradual absorption of malicious behaviour into the benign profile can be detected.

9.4 Threats to validity

Search and selection. The field moves quickly, and several of the most relevant studies appeared in 2026. Snowballing from the included studies reduces this risk without removing it.

Extraction. The entries "none identified" and "not reported" rest on the published descriptions of each study. A mechanism that exists in released code but is not described in the publication would be missed.

Classification. Model families overlap. A temporal heterogeneous model, for example, was assigned to the temporal family, and a different assignment would change individual cells of Table 10. It would not change the finding that no study evaluates adaptation across a dated shift.

Generalisability. Most evidence comes from the DARPA TC data. Conclusions drawn from these engagements may not hold for enterprise environments with different workloads, and the direction of that uncertainty is unknown.

10 Open challenges and future directions

This section sets out eight open problems that can be addressed with existing data and methods (Section 10.1) and four longer-term directions for the domain (Section 10.2). Fig. 4 summarises both.

10.1 Open problems

The synthesis points to eight open problems. Each is stated with the experiment that would address it (Table 11).

Prediction-time benchmarks. No public benchmark distributes provenance graphs truncated at decision time, so each group makes its own undocumented choice. The field needs graph-construction code that builds the graph at node creation and at fixed delays, with a leakage audit of identifiers and preprocessing. The modular evaluation framework released for eight detectors is a natural host for such a component (Bilot et al. 2026).

Operating-point transfer. Ranking transfer and operating-point transfer have not been measured separately for any GNN-based provenance detector. The experiment is to lock the threshold and calibrator on calibration days and to report, for each later period, the threshold-free metrics, the fixed-threshold metrics and the false positives per 100,000 benign nodes. Calibration error on later data should be reported alongside (Silva Filho et al. 2023).

Label-adjacency dependence. It is not known how much of the reported performance of GNN detectors derives from proximity to indicator-matched nodes. The experiment is to stratify results by hop distance from those nodes and to ablate the neighbours that carry the indicator. A benign-only objective trained without propagated labels provides the comparison.

Separating benign drift from attack drift. A drop in precision may come from new benign behaviour, and a drop in recall from new attack behaviour, but existing datasets change both at once. Controlled designs that replay a fixed campaign into later benign periods would separate the two.

Safe adaptation under scarce and delayed labels. Fine-tuning, replay and test-time adaptation have not been compared on provenance graphs against a frozen baseline. Any such comparison must include a poisoning condition because an adaptive detector learns from data the adversary can shape. Methods from continual graph learning are the obvious candidates (Yuan et al. 2026).

Transfer to unseen campaigns. Recurring campaigns reuse tooling, whereas new campaigns tend not to. A leave-one-campaign-out design measures whether a detector has learnt attack behaviour or the signature of a particular campaign. Mimicry results suggest that detectors relying on local structure will be fragile here (Goyal et al. 2023).

Simple baselines and cost. Graph models should be compared with tree ensembles on prediction-time features and with allowlists, under one protocol and with training time, memory and latency reported. If a GNN does not improve on the baseline at the locked operating point, that result is informative and should be published.

Uncertainty with few independent units. With tens of days and a handful of campaigns, confidence intervals over nodes are misleadingly narrow. Resampling by day and by campaign, multi-seed runs, and the hashing of models and thresholds before the test days are opened would make the limits of each claim explicit. General guidance on test-set reuse applies directly (Lones 2024).

Table 11 The open challenges, the question each raises and the experiment that would address it

Challenge	Question	Experiment
C1 Prediction-time benchmarks	How much signal exists when the alert must be raised?	Build graphs at node creation and at fixed delays; report performance against delay
C2 Operating-point transfer	Does a locked threshold hold its alert budget on later days?	Lock on calibration days; report ranking and fixed-threshold metrics for each later period
C3 Label-adjacency dependence	Does the model detect attacks or relearn the labelling rule?	Stratify by hop distance from indicator-matched nodes; ablate indicator-carrying neighbours
C4 Benign versus attack drift	Which shift causes which loss?	Replay a fixed campaign into later benign periods
C5 Safe adaptation	Does adaptation beat the frozen baseline without admitting poison?	Compare fine-tuning, replay and test-time adaptation, with and without poisoned updates
C6 Unseen campaigns	Does the detector generalise beyond known tooling?	Leave-one-campaign-out evaluation
C7 Baselines and cost	Is the graph model worth its cost?	Tree ensemble and allowlist under one protocol: report time, memory and latency
C8 Uncertainty	How firm is each claim?	Day-level and campaign-level resampling; multiple seeds; hashed locked artefacts

10.2 Future directions for the domain

The eight problems above can be addressed with existing data and methods. Four broader directions concern where the domain itself is heading.

Longer and more varied data. Public provenance datasets span at most a few weeks and contain few campaigns, so drift cannot be studied beyond the short term. The domain needs collections that cover months, include several campaigns attributed to the same actor and record benign activity from real users instead of scripts. Datasets that combine automated attack emulation with benign traces from volunteers and software agents are a first step (Cai et al. 2026). Cloud and container workloads, which change faster than the enterprise hosts of current benchmarks, would extend the same question to environments where it arguably matters most.

Language models and pretrained encoders. Large language models have entered provenance analysis, for example to turn flagged subgraphs into an attack narrative (Aly et al. 2025). Pretrained graph encoders are a likely next step. Both inherit the period of their training data, so the two-level evaluation applies to them without change: a frozen model first, and any update measured against it. They also introduce a leakage path of their own, because a model pretrained on public text may have seen the reports that describe the benchmark attacks.

The analyst in the loop. An operating point is a workload decision, and its cost is paid in analyst time (Alahmadi et al. 2022). Future detectors should therefore be evaluated with the people who triage their alerts, on the number of entities they must inspect and on whether the explanation offered is usable. Practitioners already rank alert volume and interpretability among their main concerns (Dong et al. 2023), and explanation methods for APT detection have been reviewed separately (Mutalib et al. 2024). How explanations behave when the detector is applied to later data has not been examined.

Robust and shared adaptation. A detector that learns after deployment learns from data that the adversary can shape, and graph-based detectors are already evaded by mimicry (Goyal et al. 2023). Adaptation has to be designed with that adversary in mind. A second question is whether organisations can adapt jointly without sharing audit logs. Federated knowledge distillation has been used to keep local intrusion detectors current on network traffic (Soltani et al. 2024); whether it carries over to provenance graphs, whose structure differs from host to host, remains an open question.

a Open problems that existing data and methods can address (experiments in Table 11)

C1 Prediction-time benchmarks How much signal exists when the alert must be raised?	C2 Operating-point transfer Does a locked threshold hold its alert budget on later days?
C3 Label-adjacency dependence Does the model detect attacks or relearn the labelling rule?	C4 Benign versus attack drift Which shift causes which loss?
C5 Safe adaptation Does adaptation beat the frozen baseline without admitting poison?	C6 Unseen campaigns Does the detector generalise beyond known tooling?
C7 Baselines and cost Is the graph model worth its cost?	C8 Uncertainty How firm is each claim?

b Longer-term directions for the domain

D1 Longer and more varied data <ul style="list-style-type: none"> • Collections that span months, not days • Several campaigns attributed to one actor • Benign activity from real users • Cloud and container workloads 	D2 Language models and pretrained encoders <ul style="list-style-type: none"> • The same two-level evaluation applies • Pretraining text as a new leakage path
D3 The analyst in the loop <ul style="list-style-type: none"> • The operating point as a workload decision • Entities inspected for each alert • Explanations tested on later data 	D4 Robust and shared adaptation <ul style="list-style-type: none"> • Adaptation designed against an adversary • Joint adaptation without sharing audit logs

Fig. 4 Open problems and future directions for the domain. **a** The eight open problems (C1–C8), which can be addressed with existing data and methods; the experiment for each is given in Table 11. **b** The four longer-term directions (D1–D4)

11 Conclusion

GNNs have become the main approach to learning from provenance graphs, and the 29 detection studies reviewed here report consistently high accuracy. This review shows that the evidence behind those reports does not yet answer the question that matters for deployment: what a detector does on data collected after it was trained.

Three findings support that conclusion. Time is used as a model input far more often than as an evaluation constraint, and the truncation of the graph at decision time is seldom reported. Thresholds are rarely locked before the test period, and no study reports how a locked operating point behaves across successive later periods. Five studies describe an adaptation mechanism, and none evaluates it across a dated shift.

The two-level definition of temporal transfer, the six-dimension taxonomy and the 21-item checklist are offered as tools for closing this gap. Level 1 separates ranking transfer from operating-point transfer for a frozen detector. Level 2 measures adaptation against that baseline. The eight open problems in Section 10 translate the gap into experiments that can be run on existing data, beginning with prediction-time graph construction and a locked-threshold study of a supervised and a benign-only GNN against a tree-ensemble baseline.

Until such studies are available, reported scores for provenance GNNs are best understood as retrospective classification results. Detection, in the sense of raising a correct alert at the time it is needed, on data the model has not seen, remains to be demonstrated.

Appendix A: Evaluation and dataset studies

Table 12 lists the six evaluation or dataset studies included in the synthesis (Section 1.1).

Table 12 The six evaluation or dataset studies included in the synthesis

Study	Kind	Contribution used in this review
Anjum et al. (2021)	Dataset analysis	Usefulness of the DARPA OpTC data for detection research
Goyal et al. (2023)	Adversarial evaluation	Mimicry attacks against provenance-graph detectors
Dong et al. (2023)	Industrial study	Practitioner concerns about provenance-based detection and response tools
Bilot et al. (2025)	Re-evaluation	Eight systems under one framework; a simple neural network outperforms them
Bilot et al. (2026)	Evaluation framework	Consistent protocols; sources of irreproducibility, including tuned thresholds and unreported variability
Cai et al. (2026)	Dataset analysis and new datasets	Limits of existing benchmarks; entity-level labels; time-weighted metrics

Statements and Declarations

Funding No funding was received for conducting this study.

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Author contributions Noor Hazlina Abdul Mutalib: Conceptualisation, Formal analysis, Visualisation, Writing – original draft. Aznul Qalid Md Sabri, Ainuddin Wahid Abdul Wahab, Erma Rahayu Mohd Faizal Abdullah and Nouar AlDahoul: Supervision, Validation, Writing – review and editing. All authors read and approved the final manuscript.

Data availability The data extracted from the 35 included studies, including the quality assessment, are available from the corresponding author on reasonable request.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Code availability Not applicable.

References

- Alahmadi BA, Axon L, Martinovic I (2022) 99% false positives: a qualitative study of SOC analysts' perspectives on security alarms. In: 31st USENIX Security Symposium (USENIX Security 22). USENIX Association, Boston, pp 2783–2800. <https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi>
- Alotaibi MH, Alhuzali A, Alhothali A, El Malhi M (2026) EA-THGN: elastic, trajectory-aware graph learning for provenance-based APT detection. Inf Sci 749:123518. <https://doi.org/10.1016/j.ins.2026.123518>

- Alsaheel A, Nan Y, Ma S, Yu L, Walkup G, Celik ZB, Zhang X, Xu D (2021) ATLAS: a sequence-based learning approach for attack investigation. In: 30th USENIX Security Symposium (USENIX Security 21). USENIX Association, pp 3005–3022. <https://www.usenix.org/conference/usenixsecurity21/presentation/alsaheel>
- Alshamrani A, Myneni S, Chowdhary A, Huang D (2019) A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. *IEEE Commun Surv Tutor* 21(2):1851–1877. <https://doi.org/10.1109/COMST.2019.2891891>
- Aly A, Iqbal S, Youssef A, Mansour E (2024) MEGR-APT: a memory-efficient APT hunting system based on attack representation learning. *IEEE Trans Inf Forensics Secur* 19:5257–5271. <https://doi.org/10.1109/TIFS.2024.3396390>
- Aly A, Mansour E, Youssef A (2025) OCR-APT: reconstructing APT stories from audit logs using subgraph anomaly detection and LLMs. In: *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp 261–275. <https://doi.org/10.1145/3719027.3765219>
- Amaru Y, Wudali PN, Elovici Y, Shabtai A (2025) RAPID: robust APT detection and investigation using context-aware deep learning. *Comput Netw* 273:111744. <https://doi.org/10.1016/j.comnet.2025.111744>
- Andresini G, Pendlebury F, Pierazzi F, Loglisci C, Appice A, Cavallaro L (2021) INSOMNIA: towards concept-drift robustness in network intrusion detection. In: *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec '21)*, pp 111–122. <https://doi.org/10.1145/3474369.3486864>
- Anjum MM, Iqbal S, Hamelin B (2021) Analyzing the usefulness of the DARPA OpTC dataset in cyber threat detection research. In: *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pp 27–32. <https://doi.org/10.1145/3450569.3463573>
- Apicella A, Isgro F, Prevete R (2025) Don't push the button! Exploring data leakage risks in machine learning and transfer learning. *Artif Intell Rev* 58(11):339. <https://doi.org/10.1007/s10462-025-11326-3>
- Ares-Robledo F, Rifà-Pous H, Clarisó R (2026) Graph neural networks for anomaly detection: a systematic review of dynamic temporal approaches. *Artif Intell Rev* 59(5):129. <https://doi.org/10.1007/s10462-026-11532-7>
- Arp D, Quiring E, Pendlebury F, Warnecke A, Pierazzi F, Wressnegger C, Cavallaro L, Rieck K (2022) Dos and don'ts of machine learning in computer security. In: 31st USENIX Security Symposium (USENIX Security 22). USENIX Association, Boston, pp 3971–3988. <https://www.usenix.org/conference/usenixsecurity22/presentation/arp>
- Axelsson S (2000) The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans Inf Syst Secur* 3(3):186–205. <https://doi.org/10.1145/357830.357849>
- Barbero F, Pendlebury F, Pierazzi F, Cavallaro L (2022) Transcending TRANSCEND: revisiting malware classification in the presence of concept drift. In: 2022 IEEE Symposium on Security and Privacy (SP), pp 805–823. <https://doi.org/10.1109/SP46214.2022.9833659>
- Bayram F, Ahmed BS, Kassler A (2022) From concept drift to model degradation: an overview on performance-aware drift detectors. *Knowl-Based Syst* 245:108632. <https://doi.org/10.1016/j.knsys.2022.108632>
- Bilot T, El Madhoun N, Al Agha K, Zouaoui A (2023) Graph neural networks for intrusion detection: a survey. *IEEE Access* 11:49114–49139. <https://doi.org/10.1109/ACCESS.2023.3275789>
- Bilot T, Jiang B, Li Z, El Madhoun N, Al Agha K, Zouaoui A, Pasquier T (2025) Sometimes simpler is better: a comprehensive analysis of state-of-the-art provenance-based intrusion detection systems. In: 34th USENIX Security Symposium (USENIX Security 25). USENIX Association, Seattle, pp 7193–7212. <https://www.usenix.org/conference/usenixsecurity25/presentation/bilot>
- Bilot T, Jiang B, Pasquier T (2026) PIDSMaker: building and evaluating provenance-based intrusion detection systems. In: *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26)*. <https://doi.org/10.48550/arXiv.2601.22983>
- Bing R, Yuan G, Zhu M, Meng F, Ma H, Qiao S (2023) Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artif Intell Rev* 56(8):8003–8042. <https://doi.org/10.1007/s10462-022-10375-2>
- Buchta R, Gkoktsis G, Heine F, Kleiner C (2024) Advanced persistent threat attack detection systems: a review of approaches, challenges, and trends. *Digit Threats Res Pract* 5(4):1–37. <https://doi.org/10.1145/3696014>
- Cai Q, Wang L, Zhu Y, Chen Z, Shen X, Li Z (2026) Building next-generation datasets for provenance-based intrusion detection. In: *Proceedings 2026 Workshop on Attack Provenance, Reasoning, and Investigation for Security in the Monitored Environment*. <https://doi.org/10.14722/prism.2026.23021>

- Chen T, Dong C, Lv M, Song Q, Liu H, Zhu T, Xu K, Chen L, Ji S, Fan Y (2022) APT-KGL: an intelligent APT detection system based on threat knowledge and heterogeneous provenance graph learning. *IEEE Trans Dependable Secure Comput.* <https://doi.org/10.1109/TDSC.2022.3229472>
- Chen Y, Ding Z, Wagner D (2023) Continuous learning for Android malware detection. In: 32nd USENIX Security Symposium (USENIX Security 23). USENIX Association, Anaheim, pp 1127–1144. <https://www.usenix.org/conference/usenixsecurity23/presentation/chen-yizheng>
- 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), pp 3533–3551. <https://doi.org/10.1109/SP54263.2024.00005>
- Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21(1):6. <https://doi.org/10.1186/s12864-019-6413-7>
- Cui M, Jiang Z, Li S, Ma C, Zhang K, Yang P, Feng H (2026) MGDA: a provenance graph-based framework for threat detection and attack scenario reconstruction. *Comput Netw* 274:111806. <https://doi.org/10.1016/j.comnet.2025.111806>
- DARPA (2020) Transparent Computing program data: Engagements 3 and 5. GitHub repository. <https://github.com/darpa-i2o/Transparent-Computing>. Accessed 9 October 2026
- Dong F, Li S, Jiang P, Li D, Wang H, Huang L, Xiao X, Chen J, Luo X, Guo Y, Chen X (2023) Are we there yet? An industrial viewpoint on provenance-based endpoint detection and response tools. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, pp 2396–2410. <https://doi.org/10.1145/3576915.3616580>
- Ekle OA, Eberle W (2024) Anomaly detection in dynamic graphs: a comprehensive survey. *ACM Trans Knowl Discov Data* 18(8):1–44. <https://doi.org/10.1145/3669906>
- Febrinanto FG, Xia F, Moore K, Thapa C, Aggarwal C (2023) Graph lifelong learning: a survey. *IEEE Comput Intell Mag* 18(1):32–51. <https://doi.org/10.1109/MCI.2022.3222049>
- Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. *ACM Comput Surv* 46(4):1–37. <https://doi.org/10.1145/2523813>
- Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol 70, pp 1263–1272. <https://proceedings.mlr.press/v70/gilmer17a.html>
- Goyal A, Han X, Wang G, Bates A (2023) Sometimes, you aren't what you do: mimicry attacks against provenance graph host intrusion detection systems. In: Proceedings 2023 Network and Distributed System Security Symposium. <https://doi.org/10.14722/ndss.2023.24207>
- Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), pp 1025–1035. <https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7e9-Abstract.html>
- Han X, Pasquier T, Bates A, Mickens J, Seltzer M (2020) Unicorn: runtime provenance-based detector for advanced persistent threats. In: Proceedings 2020 Network and Distributed System Security Symposium. <https://doi.org/10.14722/ndss.2020.24046>
- Han D, Wang Z, Chen W, Wang K, Yu R, Wang S, Zhang H, Wang Z, Jin M, Yang J, Shi X, Yin X (2023) Anomaly detection in the open world: normality shift detection, explanation, and adaptation. In: Proceedings 2023 Network and Distributed System Security Symposium. <https://doi.org/10.14722/ndss.2023.24830>
- Hou Z, Liu X, Cen Y, Dong Y, Yang H, Wang C, Tang J (2022) GraphMAE: self-supervised masked graph autoencoders. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp 594–604. <https://doi.org/10.1145/3534678.3539321>
- Hu Z, Dong Y, Wang K, Sun Y (2020) Heterogeneous graph transformer. In: Proceedings of The Web Conference 2020, pp 2704–2710. <https://doi.org/10.1145/3366423.3380027>
- Inam MA, Chen Y, Goyal A, Liu J, Mink J, Michael N, Gaur S, Bates A, Hassan WU (2023) SoK: history is a vast early warning system: auditing the provenance of system intrusions. In: 2023 IEEE Symposium on Security and Privacy (SP), pp 2620–2638. <https://doi.org/10.1109/SP46215.2023.10179405>
- Ismaeli Taba MR, Zare H, Rakhshani S (2026) Graph neural networks for network intrusion detection systems: a comprehensive survey, taxonomy, and empirical evaluation. *Artif Intell Rev.* <https://doi.org/10.1007/s10462-026-11683-7>

- Jiang B, Bilot T, El Madhoun N, Al Agha K, Zouaoui A, Iqbal S, Han X, Pasquier T (2025) ORTHRUS: achieving high quality of attribution in provenance-based intrusion detection systems. In: 34th USENIX Security Symposium (USENIX Security 25). USENIX Association, Seattle, pp 7173–7192. <https://www.usenix.org/conference/usenixsecurity25/presentation/jiang-baoxiang>
- Kapoor M, Melton J, Ridenhour M, Krishnan S, Moyer T (2021) PROV-GEM: automated provenance analysis framework using graph embeddings. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp 1720–1727. <https://doi.org/10.1109/ICMLA52953.2021.00273>
- King ST, Chen PM (2003) Backtracking intrusions. In: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, pp 223–236. <https://doi.org/10.1145/945445.945467>
- King IJ, Huang HH (2023) Euler: detecting network lateral movement via scalable temporal link prediction. *ACM Trans Priv Secur* 26(3):35. <https://doi.org/10.1145/3588771>
- Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations (ICLR 2017). <https://openreview.net/forum?id=SJU4ayYgl>
- Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R (2017) Overcoming catastrophic forgetting in neural networks. *Proc Natl Acad Sci USA* 114(13):3521–3526. <https://doi.org/10.1073/pnas.1611835114>
- Kuppa A, Le-Khac N-A (2022) Learn to adapt: robust drift detection in security domain. *Comput Electr Eng* 102:108239. <https://doi.org/10.1016/j.compeleceng.2022.108239>
- Li Z, Chen QA, Yang R, Chen Y, Ruan W (2021) Threat detection and investigation with system-level provenance graphs: a survey. *Comput Secur* 106:102282. <https://doi.org/10.1016/j.cose.2021.102282>
- Li H, Wang X, Zhang Z, Zhu W (2025) Out-of-distribution generalization on graphs: a survey. *IEEE Trans Pattern Anal Mach Intell* 47(11):10490–10512. <https://doi.org/10.1109/TPAMI.2025.3593897>
- Liu Y, Jin M, Pan S, Zhou C, Zheng Y, Xia F, Yu PS (2023) Graph self-supervised learning: a survey. *IEEE Trans Knowl Data Eng* 35(6):5879–5900. <https://doi.org/10.1109/TKDE.2022.3172903>
- Lones MA (2024) Avoiding common machine learning pitfalls. *Patterns* 5(10):101046. <https://doi.org/10.1016/j.patter.2024.101046>
- Ma X, Wu J, Xue S, Yang J, Zhou C, Sheng QZ, Xiong H, Akoglu L (2023) A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans Knowl Data Eng* 35(12):12012–12038. <https://doi.org/10.1109/TKDE.2021.3118815>
- Ma M, Zhu T, Li S, Chen T, Lv M, Weng Z, Chen G (2025) Actminer: applying causality tracking and increment aligning for graph-based threat hunting. *Knowl-Based Syst* 327:114169. <https://doi.org/10.1016/j.knosys.2025.114169>
- Manzoor E, Milajerdi SM, Akoglu L (2016) Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1035–1044. <https://doi.org/10.1145/2939672.2939783>
- Milajerdi SM, Gjomemo R, Eshete B, Sekar R, Venkatakrishnan VN (2019) HOLMES: real-time APT detection through correlation of suspicious information flows. In: 2019 IEEE Symposium on Security and Privacy (SP), pp 1137–1152. <https://doi.org/10.1109/SP.2019.00026>
- Mutalib NHA, Sabri AQM, Wahab AWA, Abdullah ERMF, AlDahoul N (2024) Explainable deep learning approach for advanced persistent threats (APTs) detection in cybersecurity: a review. *Artif Intell Rev* 57(11):297. <https://doi.org/10.1007/s10462-024-10890-4>
- Pan B, Stakhanova N, Ray S (2023) Data provenance in security and privacy. *ACM Comput Surv* 55(14s):1–35. <https://doi.org/10.1145/3593294>
- Patil R, Muneeswaran S, Sachidananda V, Gurusamy M (2023) E-Audit: distinguishing and investigating suspicious events for APTs attack detection. *J Syst Archit* 144:102988. <https://doi.org/10.1016/j.sysarc.2023.102988>
- Pendlebury F, Pierazzi F, Jordaney R, Kinder J, Cavallaro L (2019) TESSERACT: eliminating experimental bias in malware classification across space and time. In: 28th USENIX Security Symposium (USENIX Security 19). USENIX Association, Santa Clara, pp 729–746. <https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury>
- Qiao W, Feng Y, Li T, Ma Z, Shen Y, Ma J, Liu Y (2025) SLOT: provenance-driven APT detection through graph reinforcement learning. In: Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, pp 963–977. <https://doi.org/10.1145/3719027.3744788>
- Qiu X, Lv M, Zhu T, Song Q, Chen T (2026a) Provenance-based advanced persistent threat detection via holistic contrastive learning with heuristic augmentation. *J Inf Secur Appl* 100:104457. <https://doi.org/10.1016/j.jisa.2026.104457>

- Qiu X, Lv M, Zhu T, Zhang Y, Chen T (2026b) ProHunter: a comprehensive APT hunting system based on whole-system provenance. *Comput Secur* 167:104907. <https://doi.org/10.1016/j.cose.2026.104907>
- Ren J, Geng R (2025) Provenance-based APT campaigns detection via masked graph representation learning. *Comput Secur* 148:104159. <https://doi.org/10.1016/j.cose.2024.104159>
- Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* 10(3):e0118432. <https://doi.org/10.1371/journal.pone.0118432>
- Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M (2018) Modeling relational data with graph convolutional networks. In: Gangemi A et al (eds) *The semantic web. ESWC 2018. Lecture notes in computer science*, vol 10843. Springer, Cham, pp 593–607. https://doi.org/10.1007/978-3-319-93417-4_38
- Shyaa MA, Ibrahim NF, Zainol Z, Abdullah R, Anbar M, Alzubaidi L (2024) Evolving cybersecurity frontiers: a comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. *Eng Appl Artif Intell* 137:109143. <https://doi.org/10.1016/j.engappai.2024.109143>
- Silva Filho T, Song H, Perello-Nieto M, Santos-Rodriguez R, Kull M, Flach P (2023) Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Mach Learn* 112(9):3211–3260. <https://doi.org/10.1007/s10994-023-06336-7>
- Skarding J, Gabrys B, Musial K (2021) Foundations and modeling of dynamic networks using dynamic graph neural networks: a survey. *IEEE Access* 9:79143–79168. <https://doi.org/10.1109/ACCESS.2021.3082932>
- Soltani M, Khajavi K, Jafari Siavoshani M, Jahangir AH (2024) A multi-agent adaptive deep learning framework for online intrusion detection. *Cybersecurity* 7(1):9. <https://doi.org/10.1186/s42400-023-00199-0>
- Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp 305–316. <https://doi.org/10.1109/SP.2010.25>
- Stojanović B, Hofer-Schmitz K, Kleb U (2020) APT datasets and attack modeling for automated detection methods: a review. *Comput Secur* 92:101734. <https://doi.org/10.1016/j.cose.2020.101734>
- Tariq S, Baruwat Chhetri M, Nepal S, Paris C (2025) Alert fatigue in security operations centres: research challenges and opportunities. *ACM Comput Surv* 57(9):1–38. <https://doi.org/10.1145/3723158>
- Ur Rehman M, Ahmadi H, Ul Hassan W (2024) Flash: a comprehensive approach to intrusion detection via provenance graph representation learning. In: 2024 IEEE Symposium on Security and Privacy (SP), pp 3552–3570. <https://doi.org/10.1109/SP54263.2024.00139>
- Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph attention networks. In: 6th International Conference on Learning Representations (ICLR 2018). <https://openreview.net/forum?id=rJXMpikCZ>
- Waikhom L, Patgiri R (2023) A survey of graph neural networks in various learning paradigms: methods, applications, and challenges. *Artif Intell Rev* 56(7):6295–6364. <https://doi.org/10.1007/s10462-022-10321-2>
- Wang Q, Hassan WU, Li D, Jee K, Yu X, Zou K, Rhee JJ, Chen Z, Cheng W, Gunter CA, Chen H (2020) You are what you do: hunting stealthy malware via data provenance analysis. In: *Proceedings 2020 Network and Distributed System Security Symposium*. <https://doi.org/10.14722/ndss.2020.24167>
- Wang D, Shelhamer E, Liu S, Olshausen B, Darrell T (2021) Tent: fully test-time adaptation by entropy minimization. In: 9th International Conference on Learning Representations (ICLR 2021). <https://openreview.net/forum?id=uXl3bZLkr3c>
- Wang S, Wang Z, Zhou T, Sun H, Yin X, Han D, Zhang H, Shi X, Yang J (2022) THREATTRACE: detecting and tracing host-based threats in node level through provenance graph learning. *IEEE Trans Inf Forensics Secur* 17:3972–3987. <https://doi.org/10.1109/TIFS.2022.3208815>
- Wang L, Zhang X, Su H, Zhu J (2024) A comprehensive survey of continual learning: theory, method and application. *IEEE Trans Pattern Anal Mach Intell* 46(8):5362–5383. <https://doi.org/10.1109/TPAMI.2024.3367329>
- Wang L, Fang L, Hu Y (2025a) A dynamic provenance graph-based detector for advanced persistent threats. *Expert Syst Appl* 265:125877. <https://doi.org/10.1016/j.eswa.2024.125877>
- Wang Z, Luo Y, Zheng L, Chen Z, Wang S, Huang Z (2025b) In search of lost online test-time adaptation: a survey. *Int J Comput Vis* 133(3):1106–1139. <https://doi.org/10.1007/s11263-024-02213-5>
- Wang H, Chen W, Li L, Pu H, Zhang Y (2026) Dinspector: dual factor graph attention mechanism for advanced persistent threat detection. *Eng Appl Artif Intell* 167:113861. <https://doi.org/10.1016/j.engappai.2026.113861>
- Weng Z, Zhang W, Zhu T, Dou Z, Sun H, Ye Z, Tian Y (2025) RT-APT: a real-time APT anomaly detection method for large-scale provenance graph. *J Netw Comput Appl* 233:104036. <https://doi.org/10.1016/j.jnca.2024.104036>

- Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2021) A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 32(1):4–24. <https://doi.org/10.1109/TNNLS.2020.2978386>
- Wu L, Xie Y, Zhao S, Zhou P, Feng D, Wildani A, Wu Y (2025) Efficient intrusion detection via heterogeneous graph attention networks and parallel provenance analysis. *Comput Netw* 270:111552. <https://doi.org/10.1016/j.comnet.2025.111552>
- Xiong C, Zhu T, Dong W, Ruan L, Yang R, Cheng Y, Chen Y, Cheng S, Chen X (2022) Conan: a practical real-time APT detection system with high accuracy and efficiency. *IEEE Trans Dependable Secure Comput* 19(1):551–565. <https://doi.org/10.1109/TDSC.2020.2971484>
- Xu D, Ruan C, Körpeoglu E, Kumar S, Achan K (2020) Inductive representation learning on temporal graphs. In: 8th International Conference on Learning Representations (ICLR 2020). <https://openreview.net/forum?id=rJeW1yHYwH>
- Xu B, Gong Y, Geng X, Li Y, Dong C, Liu S, Liu Y, Jiang B, Lu Z (2024) ProcSAGE: an efficient host threat detection method based on graph representation learning. *Cybersecurity* 7(1):51. <https://doi.org/10.1186/s42400-024-00240-w>
- Yang F, Xu J, Xiong C, Li Z, Zhang K (2023) PROGRAPHER: an anomaly detection system based on provenance graph embedding. In: 32nd USENIX Security Symposium (USENIX Security 23). USENIX Association, Anaheim, pp 4355–4372. <https://www.usenix.org/conference/usenixsecurity23/presentation/yang-fan>
- Yuan Z, Sun Q, Zhou H, Shao M, Fu X (2025) A comprehensive survey on GNN-based anomaly detection: taxonomy, methods, and the role of large language models. *Int J Mach Learn Cybern* 16(7–8):4407–4432. <https://doi.org/10.1007/s13042-024-02516-6>
- Yuan Q, Guan S-U, Ni P, Luo T, Wong P, Chang V, Man KL (2026) Continual graph learning: a survey. *Pattern Recognit* 179:113600. <https://doi.org/10.1016/j.patcog.2026.113600>
- Yue H, Li T, Wu D, Zhang R, Yang Z (2024) Detecting APT attacks using an attack intent-driven and sequence-based learning approach. *Comput Secur* 140:103748. <https://doi.org/10.1016/j.cose.2024.103748>
- Zeng J, Wang X, Liu J, Chen Y, Liang Z, Chua T-S, Chua ZL (2022) SHADEWATCHER: recommendation-guided cyber threat analysis using system audit records. In: 2022 IEEE Symposium on Security and Privacy (SP), pp 489–506. <https://doi.org/10.1109/SP46214.2022.9833669>
- Zhang B, Gao Y, Kuang B, Yu C, Fu A, Susilo W (2025) A survey on advanced persistent threat detection: a unified framework, challenges, and countermeasures. *ACM Comput Surv* 57(3):1–36. <https://doi.org/10.1145/3700749>
- Zheng Y, Yi L, Wei Z (2025) A survey of dynamic graph neural networks. *Front Comput Sci* 19(6):196323. <https://doi.org/10.1007/s11704-024-3853-2>
- Zhong M, Lin M, Zhang C, Xu Z (2024) A survey on graph neural networks for intrusion detection systems: methods, trends and challenges. *Comput Secur* 141:103821. <https://doi.org/10.1016/j.cose.2024.103821>
- Zhu T, Yu J, Xiong C, Cheng W, Yuan Q, Ying J, Chen T, Zhang J, Lv M, Chen Y, Wang T, Fan Y (2023) APTSHIELD: a stable, efficient and real-time APT detection system for Linux hosts. *IEEE Trans Dependable Secure Comput* 20(6):5247–5264. <https://doi.org/10.1109/TDSC.2023.3243667>
- Zipperle M, Gottwalt F, Chang E, Dillon T (2023) Provenance-based intrusion detection systems: a survey. *ACM Comput Surv* 55(7):1–36. <https://doi.org/10.1145/3539605>
- Zipperle M, Zhang Y, Wang M, Chang E, Dillon T (2026) BPGV: behavioral provenance graph views to enhance anomaly detection. *Int J Inf Manag Data Insights* 6(1):100397. <https://doi.org/10.1016/j.jjime.2026.100397>