

# Advanced Persistent Threat Attack Detection Systems: A Review of Approaches, Challenges, and Trends

Robin Buchta, George Gkoktsis, Felix Heine, Carsten Kleiner

Suggested citation:

Buchta, Robin, George Gkoktsis, Felix Heine, and Carsten Kleiner. 2024. "Advanced Persistent Threat Attack Detection Systems: A Review of Approaches, Challenges, and Trends." *Digital Threats: Research and Practice* 5 (4): 1–37. <https://doi.org/10.25968/opus-3597>.

## Abstract

Advanced persistent threat (APT) attacks present a significant challenge for any organization, as they are difficult to detect due to their elusive nature and characteristics. In this article, we conduct a comprehensive literature review to investigate the various APT attack detection systems and approaches and classify them based on their threat model and detection method. Our findings reveal common obstacles in APT attack detection, such as correctly attributing anomalous behavior to APT attack activities, limited availability of public datasets and inadequate evaluation methods, challenges with detection procedures, and misinterpretation of requirements. Based on our findings, we propose a reference architecture to enhance the comparability of existing systems and provide a framework for classifying detection systems. In addition, we look in detail at the problems encountered in current evaluations and other scientific gaps, such as a neglected consideration of integrating the systems into existing security architectures and their adaptability and durability. While no one-size-fits-all solution exists for APT attack detection, this review shows that graph-based approaches hold promising potential. However, further research is required for real-world usability, considering the systems' adaptability and explainability.

Terms of use

CC BY 4.0

This document is made available under these conditions:  
**Creative Commons - CC BY - Namensnennung 4.0 International**  
For more information see:  
<https://creativecommons.org/licenses/by/4.0/deed.de>





# Advanced Persistent Threat Attack Detection Systems: A Review of Approaches, Challenges, and Trends

ROBIN BUCHTA, Institute for Applied Data Science Hannover (Data | H), Hanover, Germany

GEORGE GKOKTSIS, Fraunhofer SIT - ATHENE, Darmstadt, Germany

FELIX HEINE and CARSTEN KLEINER, Institute for Applied Data Science Hannover (Data | H), Hanover, Germany

---

Advanced persistent threat (APT) attacks present a significant challenge for any organization, as they are difficult to detect due to their elusive nature and characteristics. In this article, we conduct a comprehensive literature review to investigate the various APT attack detection systems and approaches and classify them based on their threat model and detection method. Our findings reveal common obstacles in APT attack detection, such as correctly attributing anomalous behavior to APT attack activities, limited availability of public datasets and inadequate evaluation methods, challenges with detection procedures, and misinterpretation of requirements. Based on our findings, we propose a reference architecture to enhance the comparability of existing systems and provide a framework for classifying detection systems. In addition, we look in detail at the problems encountered in current evaluations and other scientific gaps, such as a neglected consideration of integrating the systems into existing security architectures and their adaptability and durability. While no one-size-fits-all solution exists for APT attack detection, this review shows that graph-based approaches hold promising potential. However, further research is required for real-world usability, considering the systems' adaptability and explainability.

CCS Concepts: • **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; • **Computing methodologies** → Machine learning; • **General and reference** → **Surveys and overviews**;

Additional Key Words and Phrases: Cybersecurity, APT, attack detection, machine learning, artificial intelligence

## ACM Reference format:

Robin Buchta, George Gkoktsis, Felix Heine, and Carsten Kleiner. 2024. Advanced Persistent Threat Attack Detection Systems: A Review of Approaches, Challenges, and Trends. *Digit. Threat. Res. Pract.* 5, 4, Article 39 (December 2024), 37 pages. <https://doi.org/10.1145/3696014>

---

## 1 Introduction

Attack Detection is a crucial part of cyber-resiliency engineering, according to the **National Institute of Standards and Technology (NIST)** [121]. While informed by the entirety of the threat landscape, this study focuses mainly on **advanced persistent threat (APT)** and the disruption such an adversary can cause to our systems. APTs obtain and maintain a persistent presence on their target. They include espionage and sabotage

---

This work was supported by the Federal Ministry for Economics and Climate Action (BMWK) Germany under Grant 03EI4028B. Authors' Contact Information: Robin Buchta (corresponding author), Institute for Applied Data Science Hannover (Data|H), Hanover, Germany; e-mail: robin.buchta@hs-hannover.de; George Gkoktsis, Fraunhofer SIT - ATHENE, Darmstadt, Germany; e-mail: george.gkoktsis@sit.fraunhofer.de; Felix Heine, Institute for Applied Data Science Hannover (Data|H), Hanover, Germany; e-mail: felix.heine@hs-hannover.de; Carsten Kleiner, Institute for Applied Data Science Hannover (Data|H), Hanover, Germany; e-mail: carsten.kleiner@hs-hannover.de.



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

© 2024 Copyright held by the owner/author(s).

ACM 2576-5337/2024/12-ART39

<https://doi.org/10.1145/3696014>

in their motivation while having the resources and technical capability to produce real-world hazards, physical damage, or even threaten human life [6]. It is self-evident that such risks are unacceptable. An APT attack refers to a specific attack originating from an APT.

Effective detection methods protect an organization's systems and data from unauthorized access, modification, or destruction. Methods for successful detection include monitoring suspicious activity, analyzing logs and network traffic, and using security tools such as **intrusion detection systems (IDSs)** and **security information and event management (SIEM)** systems.

Early detection also helps organizations comply with regulatory requirements, minimize the damage caused by security breaches, and return to normal operations as quickly as possible.

The start of the first cyber war [95, 134, 140] highlights the urgent need for advanced protection against cyber threats, especially in critical infrastructures. The discovery of Pipedream [34, 40], a modular attack toolkit targeting industrial control systems, urges us to enhance our efforts to detect and defend against advanced threats. Its capabilities significantly improve its predecessors, and its potential impact is severe. In this generally evolving threat landscape, we expect threat actors to continuously improve their capabilities to evade existing defenses, as showcased by recent fileless malware [1] residing in memory. Their **tactics, techniques, and procedures (TTPs)** have and are expected to become substantially more complex.

The detection field utilizes multiple **machine learning (ML)** methods to enhance security through more reliable intrusion detection, leveraging abundant cybersecurity data. However, current systems face challenges in accurately countering APT activity with high efficiency and timeliness.

Our article aims to answer the following **research questions (RQs)**:

- RQ1: What are the specifics regarding detecting an APT attack?
- RQ2: How does a detection system address the particularities of an APT attack, and which existing approaches and systems are particularly suitable?
- RQ3: What criteria and datasets are currently used to evaluate the effectiveness of an APT attack detection system, and which should be used?
- RQ4: What are the existing research limitations in APT attack detection, and where is the most potential for future detection improvements?

This article contributes by (i) systematizing the literature on APT attack detection with a special focus on detection systems, (ii) providing an overview of categories, threat landscape, detection methods, and evaluation mechanisms, (iii) identifying scientific gaps in APT attack detection, (iv) assess of the current state of the art on APT attack detection, and (v) identifying the crucial components of an APT attack detection system.

The review is structured as follows: Section 2 presents the methodology used to conduct the literature review and thus forms the basis of the work. Section 3 describes the nature of APTs and classifies them. Subsequently, Section 4 describes the different detection types and learning methods. This part on general concepts for classifying APT attack detection systems ends with a description of the evaluation of such systems in Section 5. Here, we describe the evaluation metrics and deal separately with the detection granularity and the benchmark datasets. Section 6 consults the knowledge from the previous Sections and thus presents initial findings by presenting an APT attack detection system reference architecture and addressing the comparability of the work necessary for the following main part of the article. Section 7 describes the literature search results. The Section goes into more detail about APT attack detection systems and classifies them. Section 8 discusses the review's overall results, and Section 9 summarizes possible future research directions. Section 10 presents related reviews and related research in the field. The article concludes with a summary in Section 11.

Figure 1 shows a schematic overview of the main contributions and their interaction. The overview and explanation of the components are assigned to RQ 1. The literature classification answers RQ 2. The evaluation criteria addressed in RQ 3 are reflected in the basic components and the detailed analysis of the APT attack detection systems. Finally, RQ 4 is addressed in the article and results from all the outlined contributions.

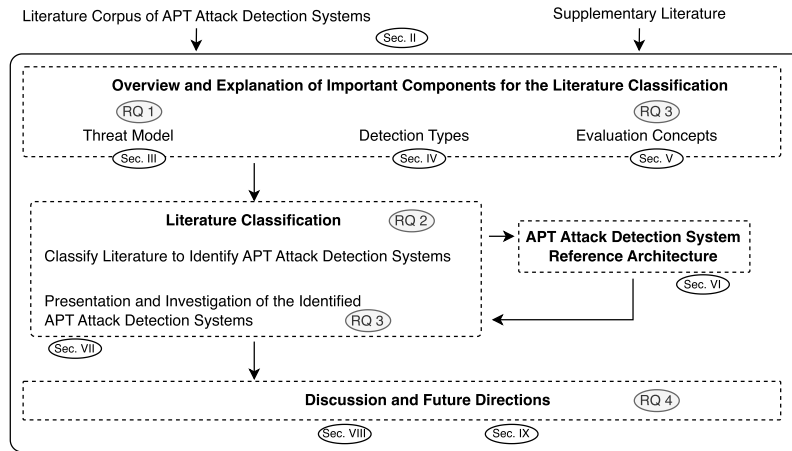


Fig. 1. Schematic overview of the article.

## 2 Methodology

We identify relevant literature through a two-step process, creating a literature corpus for further investigations. The first step follows a literature search methodology based on the **“Preferred Reporting Items for Systematic Reviews and Meta-Analyses” (PRISMA)** guidelines [114] to search the literature databases for literature on specific keywords stated below. This work is the starting point for forward and backward scanning of related works. As our review explicitly focuses on APT detection, studies that consider individual TTPs are outside the scope of the score.

The systematic literature search based on PRISMA is divided into identification and screening, accompanied by exclusion criteria. The result is an initial literature corpus from which we start the forward and backward search based on relevant works. Below are the three steps that make up our systematic search based on PRISMA with the supplementary forward and backward search:

- Identification: We formulate search strings with keywords based on the RQs. The search strings are the following:<sup>1</sup>

- “Advanced Persistent Threat” and “Detection”
- “APT” and “Detection”

We have opted for simple search terms to cover as much as possible. The search strings were used within the paper title with the help of the respective database’s advanced search function. Furthermore, we limit the selection of suitable literature databases to *IEEE Xplore* and the *ACM Digital Library* and the search space to published works from 2010 to 2022. These limitations are appropriate because we want to create an initial literature corpus. We then expanded this corpus through forward and backward searches to include a broader range of relevant literature.

- Screening: Eligibility criteria were formulated and applied to the corpus as Exclusion Criteria, as presented in the following:

- Paper is not in English,
- Paper is not accessible,
- Short paper (less than five pages),

<sup>1</sup>Note that we did not employ keywords specifically aimed at APT attack detection in our study. We made this decision in recognition that APT attack detection often encompasses the broader field of APT detection, and it is common for many authors to equate these two. It is essential to underscore that our research distinctly separates threat and attack detection domains.

- Paper is irrelevant to APT attack detection,
  - The information from the paper abstract is irrelevant to the RQs,
  - Paper is outdated (refuted or considered legacy),
  - Paper is already in the corpus (duplicate),
  - Paper employs disproven or inconsistent methodology.
- Forward and Backward Search: During screening, we marked relevant works for the backward search, and we used [online-scholar-tools<sup>2</sup>](https://www.semanticscholar.org/) for the forward search. According to the title and abstract of the respective papers, they are added to the corpus and subjected to screening and forward and backward searches. The process ends when we find no more relevant related works. We determine the relevance of a paper in the backward search based on its topic and its importance for the paper to be reviewed, and in the forward search, primarily based on the title and then the abstract.

The initial search yielded 93 papers. After screening, this number was reduced to 41. A forward and backward search, unrestricted by time or literature databases, finally increased the total to 70 papers. 13 of these papers came from other literature databases; the others were added outside the time or the keyword search. After we form the literature corpus, we analyze and categorize it in Section 7. For the classification, we examine the three areas of threats, detection methods, and evaluation to select appropriate categories. With this knowledge, we conduct a detailed examination of the works classified as APT attack detection systems and perform additional categorization. The investigation led to developing a reference architecture, which we utilize to further categorize the systems by mapping their descriptions to corresponding components. Ultimately, this knowledge enabled us to offer insights into current trends and the most promising methods for APT attack detection.

### 3 Threat Model

An actor-agnostic threat model, where attacks are computer or network misuse, categorized by broad definitions, such as *Denial of Service* or *Replay* attacks, is sufficient since, for simple attacks, it does not matter who attacks.

In contrast, APT-related attack detection and defense research axiomatically focus on an attacker-centric approach. Even though semantic inconsistencies exist around the precise definition of an APT, the term describes threat *actors* [104] with sufficient resources, persistent intent, and motivation to cause harm. It is a fast-growing subject area, not only due to the inherent difficulty of detecting and attributing APT activity but also due to the damage potential that these actors are capable of, especially in the context of cyber-physical systems and critical infrastructures.

The **Department of Defense (DoD)** of the USA, in a task force report [112], classified threat actors in a six-tier scheme; Table 1 shows the description of the tiers as a verbatim quote from the original report. Based on the current definitions, the APT would fall into Tier IV to VI. One could argue that Tier VI is beyond APTs and forms a category beyond the capacity of an APT. The same report assesses the seriousness of the threat and claims that “the cyber threat is serious, with potential consequences similar in some ways to the nuclear threat of the Cold War” [112]. This sentiment warrants investing in detecting their activity and a vigilant approach to dealing with it.

Due to the peculiarities of APT attacks and the available means, the APT uses real-world means to achieve the goal in addition to the available means in cyberspace. In addition to physical communication via flash drives, as used in the Stuxnet attack [3], insider attacks are a real threat used in the APT context [20].

For this reason, we systematize the literature according to the assumptions about the threat and the capacity of the attack detection method presented. This distinction is crucial because we also scanned partial solutions or solutions that do not meet the requirements of the established threat model. Detailed descriptions of the categories follow now.

<sup>2</sup><https://www.semanticscholar.org/> and <https://www.connectedpapers.com>

Table 1. Description of the Six-Tier Threat Scheme [112]

Tier	Description
I	Relying on others to develop the malicious code, delivery mechanisms, and execution strategy (use known exploits).
II	Practitioners with more experience, with the ability to develop their own tools (from publically known vulnerabilities).
III	Practitioners who focus on the discovery and use of unknown malicious code are adept at installing user and kernel mode rootkits, frequently use data mining tools, and target corporate executives and key users (government and industry) to steal personal and corporate data with the expressed purpose of selling the information to other criminal elements.
IV	Criminal or state actors who are organized, highly technical, proficient, well-funded professionals working in teams to discover new vulnerabilities and develop exploits.
V	State actors who create vulnerabilities through an active program to “influence” commercial products and services during design, development, or manufacturing, or with the ability to impact products while in the supply chain to enable exploitation of networks and systems of interest.
VI	States with the ability to execute full spectrum (cyber capabilities in combination with all of their military and intelligence capabilities) operations to achieve a specific outcome in political, military, economic, and so forth, domains and apply at scale.

### 3.1 Conventional Attacks

This category refers to attacks by adversaries with capabilities inferior to the APT. The defining factor is the inability to discover new vulnerabilities while being capable of developing malicious code to exploit existing ones [6]. Single-step attack models or abstract descriptions of attacks differentiated in generalized attack taxonomies fall into this category.

For this purpose, attackers use known vulnerability reports or ready-made attack toolkits to look for victims according to a *try-and-error* procedure. The attacks follow a *smash-and-grab* approach and do not attempt to act covertly [6].

### 3.2 Multi-Step Attacks

A multi-step attack, often called multi-stage, attack scenario, or attack session, describes an attack that consists of at least two atomic events [110]. Detecting multi-step attacks poses several challenges, including identifying individual steps of an attack as seemingly harmless. The interval between consecutive steps of an attack can also be very diverse, ranging from hours to days to months.

According to Table 1, we classify the threat this type of attack poses as Tier II to III. They can also not discover new vulnerabilities and rely on existing vulnerabilities and tools. They differ from conventional attacks by linking several TTPs. Multi-step attacks can take many forms but often follow a similar pattern. Not all of these steps need to occur and may be modified [6]:

- Reconnaissance: In this step, the attacker gathers information about the target system, such as its vulnerabilities, user accounts, and available resources. Reconnaissance might involve online searches, social engineering, or scan tools.
- Initial compromise: Attackers exploit vulnerabilities or trick users into providing access to the system. The intrusion may involve sending a phishing e-mail, exploiting a software vulnerability, or using a stolen password.
- Lateral movement: The entry point is often not the attacker’s target, and the attacker only uses this to exploit other opportunities to achieve their own goal. To reach the intended target, attackers spy on the target network using various methods, including those used in the initial compromise.
- Escalation of privilege: The attacker attempts to gain higher levels of access to the system, such as administrative privileges, which may involve finding and exploiting additional vulnerabilities or stealing more credentials.

- Data exfiltration/Cause damage: The attacker extracts sensitive data from the system, such as intellectual property, financial information, or personal data. The attacker may also be intent on destroying or obstructing the target and will encrypt data, change parameters, and thus attempt to reach his target. The attackers cause damage to the respective targets through malware, encryption, or other domain-specific activities.
- Covering traces: In this final stage, the attackers attempt to cover their traces by, e.g., deleting or modifying log files.

Such attack patterns form the basis of Lockheed Martin’s Cyber Kill Chain [94], MITRE’s attacker-centric ATT&CK knowledge base and model [132], and the DoD’s Diamond Model [23], to which this article attempts to adhere closely. Naik et al. [109] wrote a comparative study of these three popular attack models. In this review, our primary focus is detecting attacker activities rather than threats. Specific attack models are mentioned solely in the context of developing attack detection systems that build upon them. We adhere to the nomenclature outlined by the MITRE ATT&CK framework to describe the detected mechanisms, delineating attack steps as TTPs.

### 3.3 APTs

This study focuses on APT, so it requires meticulous attention to define it. Many researchers used different descriptions to explain what advanced, persistent, or threat means. We describe APT’s defining traits as:

*Threat.* A threat refers to individuals (e.g., persons, groups, and organizations) with the motivation, capabilities, and opportunity to engage in malicious actions. A threat’s key defining characteristic is its malicious intent, which serves as the driving force behind its motivation.

*Advanced.* An advanced threat actor is technical, proficient, and capable of discovering or introducing new vulnerabilities to a system and developing novel ways to exploit these vulnerabilities. Attackers even resort to zero-day exploits, planted insiders, or supply chain exploitation when the target’s defenses require it.

*Persistent.* A persistent threat is anyone showing the determination to attack until reaching their objective, proficient in adapting to hostile territory to remain undetected, and has the resources to evolve their TTPs to defeat defensive strategic or tactical countermeasures. These resources include a variety of assets and are not specifically limited to funds. Human capital is, for example, a highly relevant resource. A persistent threat will have access to personnel capacities for every operation they initiate. They work mainly in teams, as the scope and scale of their activity are sufficiently large. Other assets are infrastructure, time, and monetary funding.

When considering APTs, it is important to remember that they are human [104], as the term was specifically designed to distinguish them from simple malware. As Richard Bejtlich argues [16]: “Threat means the adversary is not a piece of mindless code.” Technically, there is no way to capture human creativity or imagination. Therefore, we cannot assume we can predict which attack the APT will carry out. Research often considers APT attacks as targeted multi-stage attacks [110]. However, the restriction that a multi-step attack must consist of at least two atomic events is not mandatory for an APT attack. If the attacker only needs one action to achieve his goal, they will use it. The path of least resistance, as the *Advanced* part of the term, describes their ability to use the entirety of the intrusion spectrum [104]. For example, if there is a known vulnerability with corresponding exploit material, the APT *can* and *will* utilize it to achieve its goals.

A multi-step attack can also go after one specific target. Therefore, detection systems have difficulty distinguishing between the threats behind an attack.

APTs aim to remain undetected and, therefore, employ concealment techniques [6], with one of the most commonly used being *living-off-the-land* techniques [113, 127]. These techniques involve disguising malicious activity behind seemingly harmless programs that utilize tools already on users’ computers. These tools are often part of the operating system or other user-installed software. This type of attack is particularly challenging to detect, as attackers do not typically create new malicious files on target computers, and conventional antivirus scans may not identify them.

Table 2. Comparison of Conventional, Multi-Step, and APT Attacks Based on [6]

	Conventional Attacks	Multi-Step Attacks	APT Attacks
Attacker	Mostly a single individual	Criminally motivated groups	Highly organized, sophisticated, determined, and well-resourced group
Target	Unspecified, mostly individual systems	Domain-specific organizations	Specific targets with strategic value
Purpose	Financial gains, demonstrating abilities	Financial gain, demonstrating abilities	Competitive advantages, strategic benefits
Approach	Single-run, smash-and-grab, and short-duration operation	Single-run, conspicuous and not hidden, operation lasting from hours to weeks	Repeated attempts, low-and-slow approach, adapting to resist defenses long-term, and often lying in hiding until a backdoor has strategic use

These techniques are not considered in conventional **cyber threat intelligence (CTI)** as they require a higher level of sophistication from the attacker and cannot be compared to known procedures. This fact highlights the importance of considering these advanced tactics in security measures and continuously updating the CTI to stay ahead of the evolving threat landscape [144, 166].

Nevertheless, an APT can and will use known attack patterns, such as replay attacks or injections, via known vulnerabilities. Established rule-based systems and data-driven approaches for specific TTPs are a partial solution.

In their worst manifestation, an APT can be described as a *full spectrum* adversary characterized by the DoD and sometimes called integrated threats [112]. This property means they do not seek opportunities to strike; they use political, military, intelligence, or economic resources to enable their operations.

Historical examples of APT attacks of particular interest are the Stuxnet attack [3], which was used to destroy uranium enrichment centrifuges in Natanz, Iran, Industroyer [26], which caused a blackout in Ukraine in 2016, and TRITON [39], which attacked safety instrumentation systems of the petroleum industry. Among other APT attacks [27, 38, 46, 54], these represent significant cornerstones in the evolution of the APT, especially as it moved from mostly espionage to destructive attacks.

In addition to stealing information and manipulating or even destroying targets, an APT can also aim to install a backdoor to gain easy access at strategically favorable times [72]. In addition to the different uses of TTPs, attack types differ in the areas of the attacker, target, purpose, and approach. Table 2 shows the distinction between the attack types.

#### 4 Detection Types

In the field of detection, various methodologies can be employed, with the most common approach being the differentiation of tools based on their detection objectives. The employed distinction is a refined variation of the one proposed by Khalid et al. in [72], who differentiate between signature-based and anomaly-based detection at the highest level.

- Rule-based: This is a method of detecting attacks that uses predefined rules and patterns to identify suspicious activity. These rules and patterns are usually based on knowledge of known attacks or patterns of behavior considered normal in the system. Therefore, this category is also called signature-based [72] or knowledge-based detection, which can be further divided into signature matching, state transition analysis, and expert systems [101]. Rarer manifestations of the rule-based systems model through the rules allow normal behavior and thereby detect deviations. Often also called policy-based [64]. Rule-based systems are widely prevalent in practice and integrated into firewalls and IDSs [17, 74].
- Anomaly-based: Anomaly detection identifies unusual patterns or instances in data that do not conform to expected behavior [87, 101]. A key distinction is that anomaly detection does not differentiate between known

and unknown attacks. Since anomaly detection typically relies on system behavior to detect deviations, these systems are less prevalent than rule-based systems, especially considering that a learning process must precede the ML variants [87].

- Statistic: This group holds any methods based on statistical analysis for prediction and detection. Most statistical methods for attack detection utilize probabilistic models and training data to track specific behavior [45].
- Classification: The classification group holds the methods that also use ML techniques but supervised learning in which they learn benign and malicious behavior and then label new activities according to the learned classes [8]. Additionally, multiclass labeling can achieve more precise detections if the training dataset is accordingly labeled and contains a representative sample of attacks from each class [45]. New behavior that neither resembles previous benign nor previous malignant patterns cannot be acted upon appropriately [8].
- Human Knowledge: Detection systems are assigned to the human knowledge category when they leverage strong domain expert knowledge or rely on CTI.

### Learning Methods Classification

The learning category differentiates the methods systems employ to develop their models. This category is exclusively applicable to detection techniques that utilize ML.

- Unsupervised: Unsupervised learning is a type of ML in which a model is trained to recognize patterns or relationships in a dataset without being provided labeled examples to learn from. The methods presuppose complete cluelessness over the data affiliation<sup>3</sup> [108].
- Supervised: On the other hand, supervised learning is a type of ML in which the model trains to make predictions or decisions based on labeled examples. The model learns to make predictions based on the relationships between the input data and the outputs [111].
- Semi-Supervised: Semi-supervised learning combines the two previous methods. Here, the model gets a dataset that includes labeled and unlabeled examples [138].
- One-Class Learning: One-class learning, or one-class classification, refers to learning just one class. Only one class's information (supervision) is available [65]. APT attack detection is a suitable task for one-class learning, as we cannot assume that representative APT attacks are present in the data used for learning.

## 5 Evaluation Concepts

One evaluation challenge is distinguishing between method establishment and practical application, as the challenges of datasets and establishing comparability do not exist in practical application. However, there are several challenges in the scientific development of detection systems, which we discuss below. To evaluate different APT attack detection systems, we focus on the comparison metrics, detection granularity, and datasets. When we look at APT attacks, the same difficulties occur in other domains of attack detection, e.g., network attack detection [62], especially targeting evaluation metrics and meaningful datasets; only the target is different. The evaluation target is not the detection of specific attack types, such as distinguishing whether this belongs to a Denial-of-Service attack and command and control behavior. The primary target is to increase situational awareness to the point where we can detect APT attacks indicated in unusual activity.

Whereas we only use the data type for the initial classification (Section 7.1) of the literature, the metrics, detection granularity, and datasets play a crucial role when considering the selected APT attack detection systems (Section 7.2).

<sup>3</sup>Often systems, especially in the attack detection context, that learn on normal data are referred to as unsupervised learning systems rather than one-class learning. One assumption on our part is that this is done because there is no supervision over the attacks, unlike many classification or rule-based methods.

## 5.1 Comparison Metrics

The 2003 NIST publication [97], which provides an overview of existing IDS evaluation problems, proposes the following quantitative evaluation metrics:

- Coverage can refer to vulnerability coverage or attack coverage, which uses existing lists of possible attacks or vulnerabilities to determine which are detected and which are not.
- Probability of detection and false alarms indicates the false and true alarms produced in a time frame. True alarms refer to alarms that are attributable to an attack, whereas false alarms are not attributable to an attack. It can be mapped as a **receiver operating curve (ROC-Curve)** as aggregation. Thus, the percentage of true and false alarms contrasts and represents a detection system in different operating modes.
- Resistance to attacks directed at the IDS: Under this metric, different attacks on the detection system are measured, and whether the detection system continues to operate appropriately is checked. These attacks can be, for example, obviously intentional attacks to disrupt. The traffic bandwidth also belongs to this category but indicates how much load is generally acceptable.
- Ability to correlate events describes the ability of a detection system to correlate attack events from different sources. The authors [97] do not describe a possible characteristic of the metric. Conceivable categories are (1) correlation of source data, (2) correlation of alarms, and (3) any correlation.
- Ability to identify an attack refers to how well a detection system assigns a label to the alarm. This review considers this specifically in the following Section 5.2 as detection granularity.
- Ability to determine attack success: Another distinction lies in how well a detection system can distinguish successful from unsuccessful attacks.
- Other metrics (not related to detection performance) are ease of use, maintenance, setup issues, resource requirements, availability, and support quality.

Metrics that we can calculate from a confusion matrix that compares the ground truth with the results of the recognition system, such as the **true and false positive rate (TPR, FPR)**, as well as precision, accuracy, and F1-score, have become mainly established, as is shown in many works which are part of this literature review, e.g. [58, 144, 166].

While metrics such as bandwidth are essential for production-ready systems, they are secondary to the goal of APT attack detection, as even downstream detection, not in real-time, adds significant value rather than allowing the attacker to remain unnoticed in the system for longer.

The quantitative metrics are generally only comparable if the authors use the same data. For example, the values of the false alarms metric, in particular, are highly dependent on the circumstances. Considering false positives, a cost-benefit tradeoff is often useful in deciding what risk to enter, with more false positives versus possible undetected attacks [97]. In [25], the authors present a framework for evaluation, which, among other things, aims to reduce costs and put them into an overall context. The cost calculation is based on the values from the confusion matrix.

However, cybersecurity, especially in light of APT attacks, is elusive. Of course, low FPR and high TPR are desirable, but the unit of measurement is often not defined consistently. With network IDS, these are often individual flows or packets, whereby flows represent an aggregation of packets and thus lead to changed metrics. Comparability is possible at the same granularity. Nevertheless, finding a common denominator in the context of APT attacks becomes harder. Previous work mostly refers to events delivered by the underlying data collectors, which need to be specifically defined, and where a generated alarm often includes multiple events. This property is the segue to the presented metric correlatability. In the APT attack context, the correlatability of the individual methods plays a particularly important role in reducing messages and, thus, false messages. However, the next Section 5.2 will consider how messages can look.

In addition, metrics about the timing of training and the amount of data handled are important for evaluating the overall system, especially if used in production systems. However, since the recognition performance has not yet been achieved outside laboratory conditions, these metrics are not considered in detail.

## 5.2 Detection Granularity

For comparing methods, it is important to consider the different granularities. Fine granular messages are of considerable added value for the downstream steps of the cybersecurity framework [15].

However, the APTs focus on evading detection 3.3; therefore, granularity is secondary in quality assessments; first, we need to identify misuse. Priority is the possibility for detection itself. If APT attacks can be reliably detected, granularity can be refined with auxiliary processes, such as forensic analyses or matching with related **indicator of compromise (IoC)**.

We can name the following typical granularity levels, present with and without time specification, listed from fine to coarse: IoC-based, Event, Subject (e.g., Process, File), Cluster/Substructures. Beyond that, there is only a statement over the entire system with a time indication. The granularity can take different forms with the possibility of correlations. For example, we can join several nodes to a path to represent the attack as a whole or as part of a chain, as Kairos [30] does.

Datasets determine nodes' possible granularity levels, representing entities such as entire systems, individual processes, or executables. Nevertheless, when comparing respective detection granularity between different published systems, even if the same dataset is selected, different representations can have been chosen; for instance, Kairos focuses on executables, while ThreaTrace [144] focuses on processes on the same dataset. This means that when comparing performance, such differences in the detection indicators' granularity must be considered in a differentiated manner.

## 5.3 Datasets

Researchers can compare different systems by using the same dataset and evaluation metrics, establishing comparability; however, several challenges related to datasets' availability, quality, and completeness hinder this effort.

APT attacks are specific and tailored to a particular domain, as described in Section 3.3. This characteristic complicates the generalizability of results since the public datasets usually do not reflect the specifics of the environment to be protected. Therefore, the target domain can only give a final quality statement. The important methods of adaptability and feedback loops are to improve the models. In addition, APT attacks are rare, making it difficult to obtain large amounts of representative data for evaluation.

Another challenge is that the recorded data often contains sensitive information, making it difficult to publish them. In addition, data is often synthetic and does not accurately reflect the practical applications. Existing datasets often focus on only one data type, such as network data. This lack of diverse datasets limits the generalizability of assessment results. We present the datasets used in the research below.

Myneni et al. present in their work [107] the DAPT2020 benchmark dataset for APT attack detection. The focus of the DAPT2020 dataset is on attack steps that are difficult to distinguish from normal behavior and reflect both external intrusion and internal attack steps within a system. In doing so, they compare their dataset to others, mostly network intrusion detection datasets like DARPA1998 [33] and UNSW-NB-15 [105], explicitly addressing the lack of different attack steps of an APT attack and criticizing the lack of diversity in TTPs. The comparison clearly shows that completely outdated datasets, such as DARPA1998, were included and that all comparison datasets only contain network data. DAPT2020, on the other hand, keeps host and network logs.

Apart from DAPT2020, others satisfy the requirements for a potential assessment of APT attack detection systems as follows: Datasets from the **Defense Advanced Research Projects Agency (DARPA) transparent computing (TC)** program [35], the dataset from the Cyber APT Scenarios for Enterprise Systems project,

developed in cooperation with DARPA [47], CERT insider threat test dataset [85], and LANL comprehensive, multi-source cyber-security events dataset [70].

The DARPA TC program has proclaimed the STARC project [52] to develop a TC architecture enabling real-time APT attack detection. This review's selected APT attack detection systems often use the datasets; therefore, we describe the project in more detail. To this end, the authors developed a prototype based on the Lambda architecture to enable real-time analysis and forensic activities. The authors build the implementation based on Apache Kafka and multiple endpoint sinks and sources. The data gets stored in the Common Data Model format developed for this purpose, aiming to track the flow of information. Detailed insights into this data model are given in [73] and available in the repository [9, 35]. The recorded dataset can answer detailed questions about processes and user activities. Such reference system setup and topology should be applicable across various production systems. The TC program has established five **technical areas (TAs)** to achieve this goal, each staffed by several groups. TA1 is responsible for tagging and tracking, TA2 for detection, TA3 for architecture, TA4 develops test scenarios involving APT attacks, and TA5 for evaluation. The program has completed five deployments, and the datasets from Engagement Three in 2018 and Engagement Five in 2019 are publicly available [52]. The datasets are divided according to the tagging and tracking subteams: Cadets, FiveDirections, Trace, Theia, Marple (for engagement five only), and Clearscope [35].

A significant disadvantage is that only sparse information about the systems is available. Even after close examination, it is unclear how the creators establish the normal behavior, the type of system it is supposed to represent, and whether it is synthetic or comes from a practical application. The red team (TA4) documented their actions only in the supplementary documents. The unlabeled datasets leave researchers with the arduous and error-prone task of manually identifying and labeling the attack-related data. There is more information for the trace dataset because this team was also active in TA2 and published the detection results [67].

The OpTC dataset is the latest version of a public DARPA dataset for APT attack evaluation. It has the same goal, to enable the development of APT attack detection systems, but it takes a different approach. The dataset grew from another program, CHASES [117]. The program uses a different data format, the extended Cyber Analytics Repository [7], based on MITRE's CAR model [102]. The creators divided the dataset into benign and evaluation. However, the events in evaluation still need to be labeled using the ground truth. The creators recorded one thousand hosts with Windows 10 [7]. This dataset has the advantage that an enormous amount of data is available, and the publication of a survey [7] of the dataset makes it easier for scientists to get started.

Carnegie Mellon University Computer Emergency Response Team [137] is developing several datasets, including the Insider Threat Test dataset [85]. It is entirely synthetic, which makes it not representative of practical applications, but it allows it to be easily accessible to the public. It includes many data source types, such as system logs, network traffic, e-mail messages, and user activity logs. The creators also labeled the dataset, allowing researchers to evaluate their systems' performance against known attacks [51]. The dataset does not target APT attacks. However, due to the difficulty of detecting an insider attack, this challenge shares many similarities with APT activity and is potentially useful in APT detection research.

The **Los Alamos National Laboratory (LANL)** Comprehensive, Multi-Source Cyber-Security Events dataset is a real-world dataset collected by monitoring proprietary systems over 58 days. The creators recorded various data types, including network traffic, system logs, active directory logs, and authentication information. The creators also performed attacks, which are included in the dataset and marked as such. These executed attacks test the internal systems; they do not simulate an APT attack [71]. Since it reflects a real-world production system, this dataset can also offer valuable insights to researchers for APT attack detection.

In addition, authors of detection systems' articles often use private datasets that are subsequently not published for various reasons. However, they occasionally describe how they recorded the data. CamFlow [116] has especially gained popularity as an open-source provenance tracker.

Computer system behavior inherently has a graph structure, mostly in the form of provenance graphs, representing relationships between, e.g., subsystems, users, and events [30, 52, 144]. Events may exhibit various

features, such as inter-process communication, file creation, or receipt of network packets. They also consist of multiple data points representing a particular system or component and the interactions and events between them. These data points are potential nodes in a graph, and the interactions and events are potential edges between the nodes.

Overall, the graph structure provides an intuitive and efficient representation of log data's complex and dynamic relationships because a log entry indicates what was done by whom to whom [139].

Different challenges exist for trackers, especially when textual logs are present. Nevertheless, there are approaches to analyzing a log line's dynamic and static parts. In most cases, the dynamic parts represent the nodes, and the static parts represent the edges. Drain [61] is a tool researchers often use for such purposes.

Lastly, we would like to raise awareness of using cyber ranges in attack detection. A cyber range is a testbed to simulate cyber attacks and allow cybersecurity employees to practice and improve their skills in a safe and controlled environment. It can also be used for research, testing, and competitions [160]. A survey [160] proposes a taxonomy to classify the various cyber ranges. In the chapter in which the authors describe future trends and directions, automatic detection and the evaluation of detection measures are already considered. One cyber range that is publicly available under the MIT license is Masaryk University's open-source platform, KYPO [141, 142]. It is actively maintained and can be used for scientific purposes. Cyber ranges and testbeds are mostly used in in-house developments and are not yet widespread for attack detection system evaluation in the public scientific community.

#### 5.4 Data Type Differentiation of the Literature

In our review, the data type category distinguishes the detection systems based on their input sources and explicitly distinguishes whether the data is in a graph representation. The classification refers to the data sources used and does not consider the theoretically possible amount of data. The data is interchangeable and universally applicable in most methods that learn a model.

- Network: The network data includes any communication of the devices. This data input can be recorded on individual hosts, firewalls, or centrally at network routers or switches. Different sources also offer different levels of granularity (e.g., pcap or netflow), which are not further differentiated in this comparison.
- Host: All host log information is summarized under host data. Typically, these are system and audit logs.
- Other: Other input data can be, e.g., the separately stored application logs, as well as the outputs of firewalls and others.
- Graph: The distinction by graph does not indicate the type of data but the structure. A graph consists of edges and nodes, and if necessary, with attributes at edges and nodes. It is important to emphasize that the graph structure can be created both in the data collection phase and as preprocessing. The distinction here refers only to whether a graph is used or not.

### 6 APT Attack Detection System Reference Architecture

To our knowledge, this is the first reference architecture for an APT attack detection system. Its development stems from analyzing the existing proposed systems, presented in the next Section 7, and intends to assist in achieving comparability between proposed systems. With a uniform overarching design, metrics can be derived and used to compare future systems.

#### 6.1 General Design Principles

First, the general design principles for a detection system are outlined based on how APT attacks and the characteristics of its offensive behavior. Then, we describe the building blocks of the reference architecture.

- Comprehensive Coverage: The system must utilize various and diverse data sources, such as network traffic, system activity, software, and application activity, in an integrated manner to detect APT activities. We attribute this requirement primarily to the stealth and individual nature of the attacks, as explained in Section 3.
- Advanced Analysis: The system must resort to higher-level algorithmic means, as APT attacks are long-lasting, and their TTPs may vary based on their target. Thus, it must be possible to make links over a long period. This conclusion is also based on the characteristics of an APT attack; see Section 3.
- Integration: Besides advanced methods to cover the specifics of an APT attack, seamless integration with existing security tool stacks is imperative. Which follows an defense in depth methodology [10, 22]. This requirement originated from the fact that the targeted and adapted properties of the APTs imply that the adversary adapts and actively tries to evade any existing defenses and countermeasures. The path of least resistance will always be followed once found [10].
- Continuous Monitoring: The collectors must be able to continuously monitor the entire system to detect APT activities as early as possible. Without data collection, detection is not possible [10, 22].
- Adaptability: The APT attack detection system must evolve with the overall system. Updates and system extensions must not cause permanent alarms and must be considered in the analysis by the detection system for correct detection [10, 22].

Through the comprehensive monitoring of the entire system landscape, so-called defense in breadth [10], and the combination of existing detection methods with the expansion of advanced methods (defense in depth) [10], it is possible to detect APT attack activities [3]. The hints and evidence found can be combined to reconstruct an overview of the attack if not acted upon early [22].

## 6.2 Core Components of an APT Attack Detection System

*Data Collection.* This is a logical component responsible for the basis of detection. It is responsible for gathering data from different sources. There can be different collectors for the respective sources or a bundled one, e.g., in the form of a provenance tracker [10, 52, 67]. The data can be delivered in its raw form to an additional processing component, where the data is further processed. This processing can be the extraction of individual features or the transfer into a graph format. With powerful computers, this can also be done directly when the data is generated. Depending upon structure, the data of the individual systems must be enriched with meta-information so that the origin, with the view of the entire system, is still adjustable. The data collection component's architecture depends on the domain to be considered.

*Data Storage and Management.* The next indispensable component is data storage and management. It is responsible for ensuring that the data is accessible to the detection systems and maintains the data in an appropriate form and quantity so that detection engines properly consume them [10, 52, 67]. This component is mainly responsible for the system's longevity; since memory is a limiting resource, retention and reduction mechanisms are located here, enabling continuous operation and having little or no impact on recognition [58, 166]. The different manifestations of the systems classified as APT attack detection systems are contrasted in Table 6. This component also stores intermediate or partial results for further extended analyses. Activities reported as suspicious by the firewall can, e.g., be used as input in addition to raw data.

*Advanced Analytics.* The system should be able to make deep links between different heterogeneous data points over time. In this case, the criterion of deep links is based on the possible long period of APT attacks, the stealthiness, and the combination of different procedures on different subsystems. All these individual indications, or small pieces of evidence, must be considered together [10, 52, 67]. Alternatively, it should provide comparably advanced analytics that meets the requirements of APT attack detection, as described in Section 3. These can be different externally, as the analysis has shown. However, there is a trend toward graph-based anomaly detection

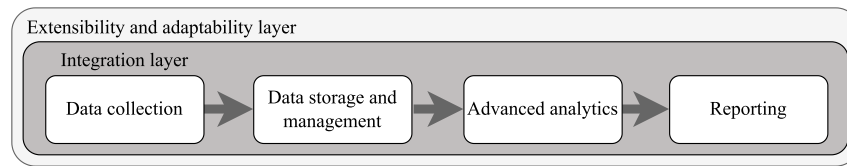


Fig. 2. Overview of the reference architecture for an APT attack detection system.

systems that attempt to form a normal behavior model and detect deviations, which is discernible; we listed examples in Section 7.2.

*Integration Layer.* The integration layer describes connections between the individual components, which can also exist in several forms. For example, the layer could combine advanced analytics with existing and conventional methods. However, the integration layer’s purpose can be reporting or visualization [144]. Likewise, other resources, such as CTI, can be brought here for enrichment to create more meaningful reports. This layer provides plenty of room for customization.

*Reporting.* The reporting component is essential as it provides an interface, the gateway, to the outside. Here, the results are displayed and, if necessary, used as a tool for defense through appropriate visualizations and additional information. With reporting, the detection responsibility ends. In addition to reporting individual incidents, statistics, metrics, and key performance indicators about the state of specific areas or the overall system can be derived and presented. Any information that increases situational awareness is appropriate [22, 115]. A further component could be triggered that enables appropriately controlled countermeasures, whether automatic, semi-automatic, or manual; this is outside the scope of this work. However, the granularity of the detection results is crucial for the success of the further steps. The more precise, and thus the higher the situational awareness of the current situation, the better countermeasures can be taken. For example, it is a decisive advantage if provenance graphs are used, as this data structure can be used for backtracking once a node has been identified as suspicious. Backtracking requires detection accuracy at a fine-grained level, e.g., node or event level, as in Kairos [30].

*Extensibility and Adaptability Layer.* A critical but often marginally considered component is the one responsible for customizing and extending the detection system. The component must be able to adapt the procedures used to new circumstances, operator feedback, or change of normal behavior [10, 67, 166]. Different possibilities are conceivable, from simple blocklists and allowlists over adding rules or models to complex retraining strategies, like periodically retraining the model, or adaptive learning, like reinforcement learning [32]. A system that theoretically detects APT attacks but suffers from threat fatigue is only a short-term solution.

We depict a graphical representation of the components and their relationships in Figure 2. It shows a data flow from collection to storage, analysis, and reporting. The integration layer wraps all four core components and ensures a flexible exchange between these components and external components. Another all-encompassing layer forms the extension and adaptability layer. Each component must evolve parallel with the system to be malleable to change.

## 7 Review and Analysis of Existing Literature

The following presents the analysis of the literature corpus (Section 7.1). Subsequently, selected works, which, according to our research, may be able to detect APT attacks, are presented in more detail and further classified, see Section 7.2.

## 7.1 Literature Categorization

The initial search yielded 93 works. After applying our exclusion criteria, we reduced this number to 41. Through iterative forward and backward citation searches, we added 29 more papers, resulting in a final literature corpus of 70 works.

The first systematization involved secluding the detection systems and classifying them further along the chosen delimitation. Sixty papers propose a detection system or approach for possible APT attack detection.

The remaining ten papers, which come from the literature search, are surveys, reviews, or discussions on various related topics around APT detection and APT attack detection, which we cannot combine along the selected dimensions: Identifying challenges through interviews [2], discussion about artificial immune systems [43], discussion of the lack of interpretability of DL models in the context of APT detection [56], evaluation of graph AI approaches for APT detection [18], discussion of **natural language processing (NLP)** approaches for APT detection [135], survey of APT detection systems [72], an overview of previous detection mechanisms [120], a provenance graph tracking system, [67], examination of previous APT IoCs [11] and game theoretic approach [154].

Tables 3 and 4 show the results of the literature classification. It is noticeable that only 18 papers fall into the APT category. Five of them, [24], [36], [96], [131], and [133], describe frameworks or meta-models that are theoretically well-suited but do not provide a concrete instance as a system and thus are not directly applicable. This result hints toward the difficulty of detection of such attacks. Often, assumptions are made that do not match our definition of an APT attack, e.g., incorporating human knowledge or CTI in the architecture of the detection system, which introduces the epistemic uncertainty of the availability of either in practical application integration. The assumption, however, that the APT attack can be detected only in network traffic or malware IoCs is fairly myopic, as the attacker using a specific attack pattern may be valid for many attacks but is unsubstantiated when considering APT attacks.

On the other hand, while assessing the suitability of the methodologies, it can be stated, based on the literature classification, that rule-based, statistical, and classification approaches do not offer sufficient attack recognition performance. For rule-based systems, the human factor usually decides, which limits the recognition performance to prior knowledge reflected in the rule base. Statistical approaches use past examples to calculate how to detect the attack or normal behavior, which only partially recognizes these approaches for APT attack detection. The classification approach works primarily with previous attacks that do not meet the requirements. This result supports the analysis of detection capabilities performed in Section 4.

Another observation is that many papers deal with multiple data sources in the form of a graph. The data resources are interchangeable in many cases. One trend we observed is that graph-based methods and methods using one-class learning anomaly detection are seemingly the most promising detection methods capable of detecting APT attacks.

Some papers do not explicitly focus on APT attacks, but the methods used to detect direct attacks or outliers could also detect these. Taking into consideration the challenges and particularities of APT attack detection [22], what is essential is the ability to detect unknown attacks. A second important factor is time, which results from an APT attacker's low and slow approach. This is because time-window-based approaches have limited predictive power in terms of APT attack detection. For example, the **Graph Neural Network (GNN)** approaches, which consider the neighborhood of an entity independent of a time constraint, are good filters only to provide the relevant information for a statement about the status of this entity. This property is not directly visible in the categorization performed, and we will add it later in Section 7.4 in the detailed consideration of APT attack detection systems.

However, many systems could not be classified as APT attack detection systems, as they do not meet the requirements of APT activities. Furthermore, the data must be detailed enough to follow possible traces of the attacker. Lastly, we cannot assume that sample attacks exist to train models or create rules. Lastly, cannot assume

Table 3. Classification of the Identified Literature (Part One)

Paper	Threat Model			Data Type				Detection Type					Learning Method			
	Conventional	Multi-Step	APT	Network	Host	Other	Graph	Rule-based	Anomaly-based	Statistic	Classification	Human Knowledge	Unsupervised	Supervised	Semi-Supervised	One-Class
Manzoor et al. (StreamSpot) [93]			x		x		x		x							x
Luh et al. (TAON) [88]	x			x	x	x	x	x				x				
Wen et al. [148]		x		x	x	x		x				x				
Shan-Shan and Ya-Bin [126]	x			x				x				x				
Wen et al. [149]	x			x	x					x	x	x		x		
Yu et al. (NetWalk) [164]			x			x	x		x				x			
Li et al. [82]	x				x						x				x	
Liu et al. (Log2vec) [86]			x	x	x		x		x				x			
Zou et al. [171]		x		x	x			x				x				
Laurenza et al. [78]	x					x					x				x	
Alageel and Maffeis (Hawk-Eye) [4]	x			x							x			x		
Eke et al. [42]	x			x							x			x		
Wilkens et al. [150]		x		x				x				x				
Wu et al. [152]	x			x			x	x				x				
MacDermott et al. [91]	x			x					x	x	x		x	x	x	x
Ayoade et al. [13]		x			x				x			x		x		
Chandran et al. [28]	x				x						x			x		
Wang et al. [145]		x		x			x		x						x	
Xuwei et al. [159]	x					x				x		x				
Yang et al. (Poirot) [162]	x					x				x				x		
Debatty et al. [37]	x			x			x		x				x			
Yu et al. [163]	x				x			x				x				
Xiong et al. (Conan) [157]		x			x	x		x				x				
Lee et al. [81]	x			x						x						x
Mangalanny and Ramli [92]	x			x					x			x				x
Ghafir et al. [49]	x					x				x	x	x			x	
Ma et al. [90]	x			x			x		x			x				
Shan-Shan and Ya-Bin [125]	x				x					x	x	x		x		
Giura and Wang [50]		x		x	x	x		x				x				
Brogi and Tong (TerminAPTor) [21]	x					x		x				x				

that sample attacks exist to train models create rules. The extended analytics component was the main focus of the classification since the other components and layers are usually easier to replace or supplement.

To present the methods specifically, we omitted the individual descriptions of the methods that we do not classify as possible APT attack detection systems.

Table 4. Classification of the Identified Literature (Part Two)

Paper	Threat Model			Data Type				Detection Type					Learning Method			
	Conventional	Multi-Step	APT	Network	Host	Other	Graph	Rule-based	Anomaly-based	Statistic	Classification	Human Knowledge	Unsupervised	Supervised	Semi-Supervised	One-Class
Mees and Debatty [96]			x	x	x	x	x		x	x	x		x	x	x	x
Zimba et al. [169]		x		x			x	x							x	
Zimba et al. [168]	x			x	x	x	x				x			x		
Su et al. [133]			x	x	x	x		x	x	x	x	x	x	x	x	x
Sree et al. [131]			x	x	x	x	x	x	x	x	x	x	x	x	x	x
Wang et al. (ThreaTrace) [144]			x	x	x	x	x		x		x					x
Shu et al. [130]	x					x			x			x		x		
Sexton et al. [124]		x		x	x	x			x	x						
Paudel and Huang (Pikachu) [118]			x	x	x	x	x		x							x
Mohamed and Belaton [103]	x					x				x		x				
Li et al. [84]			x	x	x		x		x		x					x
Kurniawan et al. (Krystal) [76]		x		x	x	x	x	x				x				
Jiang et al. [68]	x			x	x	x	x				x			x		
Han et al. (Unicorn) [58]			x	x	x	x	x		x							x
Han et al. (FRAPuccino) [59]			x	x	x	x	x		x							x
Ghafir et al. [48]	x			x				x				x				
Wei et al. (DeepHunter) [146]	x			x	x	x	x				x	x		x		
Du et al. (DeepLog) [41]	x			x	x	x			x							x
Xuan and Dao [158]	x			x							x			x		
de Vries et al. [36]			x	x	x	x		x	x	x	x	x	x	x	x	x
Bowman et al. [19]	x			x	x	x	x		x					x		
Barre et al. [14]	x			x	x	x	x				x			x		
Xie et al. (Pagoda) [155]			x	x	x	x	x	x	x							x
Zengy et al. (ShadeWatcher) [166]			x	x	x	x	x		x							x
Zhang et al. [167]	x			x	x	x			x					x		
Cam [24]			x	x	x	x	x		x	x	x		x	x	x	x
Hossain et al. (Sleuth) [64]			x	x	x	x	x	x				x				
Alsaheel et al. (Atlas) [5]	x			x	x	x	x				x			x		
Wang et al. (Provdetector) [143]			x	x	x	x	x		x							x
Cheng et al. (Kairos) [30]			x	x	x	x	x		x		x					x

### 7.2 APT Attack Detection Systems Description

This Section describes the systems from the literature research that can detect APT attacks according to our research. However, we do not consider the frameworks, meta-models, surveys, or architectures their authors have not implemented in a system. Therefore, [24], [36], [96], [131], and [133] remain disregarded even if classified as an APT attack detection system. We briefly explain the method employed for each solution, compare various properties to analyze them in-depth, and provide more context in Tables 3 and 4, as well as in Section 7.3.

Manzoor, Milajerdi, and Akoglu [93] present StreamSpot, a method for APT attack detection on host data. The method is based on a graph representation. Since most anomaly detection systems are based on static graphs, the authors have had to incorporate update and streaming functions. For efficient anomaly detection, the environment of the nodes is analyzed and transformed by locality-sensitive hashing [66] into another data structure on which cosine similarity is applied to determine a score for anomaly detection. StreamSpot needs benign data for the initialization of the clusters.

Yu et al. present the system NetWalk [164]. NetWalk processes graph data dynamically and performs anomaly detection on them. The method builds on a random walk algorithm for the neighborhood environment of each node. This vertex-formed information is then processed in a deep auto-encoder with clique embeddings to be fed to a simple clustering. The clique embeddings are a custom development of the authors, inspired by the skip-gram architecture [99] and aiming to learn a vector representation using a stream of walks, where the pairwise distance of all vertices in a walk is tried to be reduced. Therefore, the reconstruction error serves as a global clustering used for anomaly detection, where the anomaly score for each node is the lowest distance to a cluster. The method has not yet been evaluated for APT attack detection, but the authors cite that as a possible use case.

In [86], Liu et al. present Log2vec, a system for detecting insider and APT attacks. The approach is based on rules that build a graph out of logs, focusing on mapping user behavior. Graph embeddings are then created using a customized random walk algorithm and Word2vec [98]. These are clustered using pair-wise cosine distance comparisons. The assumption behind detection is that small clusters are conspicuous and deviate strongly from normal behavior in the similarity comparison due to the creation of the node embeddings. The system uses rules only for creating the graph; therefore, the approach is not rule-based in our terminology.

Wang et al. present ThreaTrace [144], a host-based APT attack detection system that uses provenance data as its basis. For the detection, the semantically rich graph structure is used as input for a GNN based on GraphSAGE [55]. The training uses benign behavior data, classified according to their type. A multi-modal procedure increases the confidence in the classification by forming several submodels in which the nodes can each be uniquely classified. The detection is done in a real-time, high-performance environment<sup>4</sup> where new nodes are connected to previous nodes and edges until a boundary is reached. After classification, if a node is not correctly classified in any submodel, it is malicious. Then, the candidate is held for a period in case it is detected as benign within another submodel. The nodes are then reported and allow for tracing, as both these and their neighborhood are reported to the operator. The classification model is supervised learned, where the information itself is contained in the raw data. Thus, no labeling is necessary. In addition, the training assumes that the training data contains only normal behavior, so the final model is derived from one-class learning.

Paudel and Huang present Pikachu [118], a one-class learned, temporal walk-based dynamic network embedding method for APT attack detection. They consider a streaming graph as a sequence of graph snapshots of fixed length (e.g., one minute or a day). The neighborhood of nodes is determined by temporal walk within a snapshot and stored by a skip-gram-based method. The temporal walks are based on another work [80]. They describe a section of a graph at a certain point in time. Here, the temporal aspects are considered. For long-term analysis across snapshots, a **gated recurrent unit (GRU)**-based autoencoder is used. Anomaly detection is done by forecasting the probabilities of each edge between two nodes at a time.

A hierarchical approach [84] to APT attack detection is presented by Li et al. They call it holistic because they include both host and network data and are hierarchical because they look at the data sequentially: (1) the host data; (2) the network data; and (3) mapping to APT attack steps. The method is based on constructing a graph, applying GNN methods, and then performing anomaly detection using an autoencoder and the reconstruction loss to predict unusual edges. The method also uses negative sampling based on the classification in a one-class learning setting. Any benign edge is negated, and thus, anomalous behavior is learned. The model is

<sup>4</sup>The authors do not describe it in their paper, but from the published code [136], we can see that they have built the environment using GraphChi.

supervised and trained from the normal data plus negative samples generated from the normal behavior, and therefore, we classified it as one-class learning. The method was not evaluated holistically but separately on different datasets.

Han et al. present, as they proclaim, a provenance-based detector for APT attacks [58]. The system is called Unicorn and is based on three open-source tools: CamFlow [116],<sup>5</sup> GraphChi [77], and HistoSketch [161]. The idea behind Unicorn is that CamFlow, which is also interchangeable, is used to generate provenance graph data, which is then processed with GraphChi into HistoSketches, where the neighborhood of a node is described. A **Weisfeil-Lehman (WL)** subtree graph kernel algorithm describes the neighborhood based on the one-dimensional WL test of isomorphism [147]. The sketches are of fixed length and are used as input for a simple comparison. It is not an anomaly if the newly generated sketch is similar enough to a previously learned sketch. Alarms are made at the sketch level, which involves a larger neighborhood.

The same team previously developed a different system called FRAPPuccino for fault detection [59]. They use CamFlow for provenance graph data generation and GraphChi for processing based on sliding windows. Normal behavior is mapped by label propagation and new feature vectors, and abnormal behavior is detected by clustering. This work serves as the predecessor of Unicorn and, therefore, shares most of the features; the big difference is in the creation and utilization of the neighborhood. FRAPPuccino uses label propagation to describe a node with its neighborhood and then  $k$ -means clustering to identify unusual nodes.

Some authors from Unicorn and FRAPPuccino are also involved in Kairos [30]. Kairos is an anomaly-based APT attack detection system that uses deep graph learning and community discovery through causal dependencies in a provenance graph. It aims to learn a graph's temporal and structural relationships with the **temporal graph network (TGN)** [122]. Kairos has the following key features: Streaming fashion: It can detect attacks in real-time with an encoder-decoder architecture for link type reconstruction for anomaly detection. The encoder is a GNN based on the UniMP [129] architecture; the decoder is a multi-layer perceptron. In addition, a node state model is trained with a recurrent neural network, the GRU [31]. The authors address the disadvantage of Unicorn in providing only coarse-granular reports and ThreaTrace in providing node-level reports. It provides attack scenario graphs that are fine-grained and visualize initial backtracking steps. The method used first builds a provenance graph of system activity. This graph captures the causal relationships between different system events. Then, the TGN model is learned. This model is used to predict the edge type of each edge in the graph. Kairos uses a time window-based anomaly detection mechanism. It maintains a queue of time windows, where each time window contains all system events within a specific time. Then, it periodically scans the queue and identifies time windows containing many anomalous edges. These time windows are flagged as suspicious. Retraining is also supported.

Pagoda [155], an APT attack recognition system by Xie et al., uses provenance graph data based on a trained set of rules considering individual provenance paths. Creating the ruleset is similar to that of a predecessor system PIDAS [156] from the authors. The rules are derived from benign behavior through dependency relationships. The provenance data provide the basis, and if patterns occur more often than a predefined threshold, they are included as rules. In the case of suspicious provenance paths, the entire graph is flagged, thus increasing the detection robustness. The system features an end-to-end pipeline that considers data acquisition, pruning, storage, rule retraining, and forensic analysis. A set of rules forms the normal behavior model; thus, deviations can be detected. This property makes this work unique because it is a one-class learning rule-based system. In the system, the provenance data collector is PASS [106]. The approach of learning the rules has already been studied more deeply in network intrusion detection and with more complex ML methods rather than with simpler threshold techniques [12].

Zeng et al. introduce ShadeWatcher [166]. The system is based on a recommender system that predicts how unusual an edge in the recorded graph is. These results are used for APT attack detection. Unusual edges are

<sup>5</sup>A subset of the authors was also involved in the development of CamFlow.

classified as an anomaly. System audit data is used as input, which is transformed into user-item relationships. The entire graph is divided into subgraphs based on behavior patterns. For this purpose, a previously developed open-source tool, Watson [165], is used. This tool serves for a more targeted detection and reduces the amount of data processed.

In [64], Hossain et al. present Sleuth. Sleuth is a real-time reconstruction test tool for attack scenarios. The tool works with commercial-off-the-shelf audit logs and creates a provenance graph. Unfortunately, the authors do not give further information about the creation of the graphs. The idea is based on the principles of Backtracker [75], which are applied to streaming data. Individual nodes are assigned tags that evaluate trustworthiness and confidentiality. With tag propagation, node tags are adapted. Initial tags are assigned to nodes by a predefined policy. Tag propagation expands the tag base so that critical states can be detected this way. The initial tagging is based on expert knowledge. Policies achieve trust detection; for instance, a high-trust tag node may not execute a low-trust tag node.

Wang et al. present Provdetector [143], a system designed to identify stealthy malware. OS-level provenance data is used for this purpose. These are transformed into a graph, from which provenance paths are cut. Benign data embeddings are learned through Doc2Vec [79] and then used with a novelty density algorithm [123] for outlier detection based on the local outlier factor. The idea is if, in the embedding space, a node has a lower locale density than its neighbors. The path selection is based on a rareness-based method, which identifies the  $k$  most unusual (suspicious) paths since each path considered would exceed the computing power. The temporal component is considered since only temporally advanced edges are relevant for the origin of a node. A regularity score [60] from previous work by the authors is used to identify the rarest paths. Not every possible provenance path is analyzed, only preselected, during attack detection. Predefined metrics do this preselection and provide, as mentioned, only the *top k* most suspicious paths.

### 7.3 APT Attack Detection System Classification

In addition to a brief description of the methods behind the systems, they are further categorized in this Subsection. Table 5 refers to the datasets used, the evaluation metrics, and the systems with which the authors compared themselves.

It is noticeable that there is no fixed comparison, nor is there a uniform benchmark dataset, as we have argued in Section 5. The DARPA TC datasets stand out in the evaluation because they are strongly represented in Table 5 and were also often considered in the rest of the literature. The evaluation metrics are primarily the conventional metrics used in ML derived from the confusion matrix. Sleuth's [64] metrics, where the authors were also part of the DARPA TC program, are particularly interesting, which we have already highlighted as desirable in the evaluation Section 5. The authors do not rely exclusively on conventional metrics but also consider the quality from the attacker's perspective.

What is striking about comparing the systems is that there is minimal overlap. Some systems partially compare with select predecessors, such as ThreaTrace with Log2vec or Pikachu with Netwalk. StreamSpot is the most frequently compared system thrice with other systems. This distinction could be related to the fact that it is one of the first methods on graph data, and the authors published their dataset, so comparability is easy to establish. We did not include the dataset in the subsection on datasets 5.3 because the dataset does not meet our requirements for an APT attack evaluation dataset. The dataset is publicly available [57] in a highly preprocessed form, with little information about the record. It consists of five graphs, each recording browsing behavior like ordinary browsing. One record contains a drive-by download attack [57, 93].

### 7.4 Integration into the Reference Architecture

For a deeper analysis of APT attack detection systems, we use the proposed reference architecture, see Section 6 and compare the system components in tabular form. Table 6 shows the overview. The specifications described by

Table 5. First Part of the Deeper Classification of APT Attack Detection Systems

Paper	Datasets	Detection Evaluation Metrics	Comparison
StreamSpot [93]	Own (published) [57]	Precision, Recall, AUC, and ROC-Curve	n/a
NetWalk [164]	UCI Messages, Digg, arXiv, and DBLP	AUC, Accuracy	Node2vec [53], DeepWalk [119]
Log2vec [86]	CERT Insider Threat Test Dataset [85], LANL [70]	AUC	StreamSpot [93], Tiesias [128]
ThreaTrace [144]	DARPA TC3 and 5 [35], StreamSpot [57], and Unicorns data	Precision, Recall, Accuracy, and F-score	Log2vec [86], DeepLog [41], LogRobust [167], and LogGAN [153]
Pikachu [118]	LANL [70], OpTC [47]	AUC	NetWalk [164], SedanSpot [44], and F-Fade [29]
Li et al. [84]	StreamSpot [57], LANL [70]	Precision, Recall, Accuracy, F-Score, AUC, and ROC	StreamSpot [57], Unicorn [58]
Unicorn [58]	StreamSpot [57], DARPA TC3 [35] (Cadets, Clearscope, and Theia), Own	Precision, Recall, Accuracy, and F-Score	StreamSpot [57]
FRAPPuccino [59]	Own	n/a	n/a
Pagoda [155]	Own	Detection-time, Detection-rate, and False alarm rate	PIDAS [156]
ShadeWatcher [166]	Trace [35]	Precision, Recall, Accuracy, F-Score, AUC, and ROC	n/a
Sleuth [64]	DARPA TC [35] (involved during execution)	Right and wrong identified and not considered rate, Detected campaign	Different versions
Provdetector [143]	Own	Precision, Recall, and F-Score	Self-created baselines
Kairos [30]	Subset of DARPA TC3 and 5 [35], StreamSpot [57]	Precision, Recall, Accuracy, and AUC	Unicorn [58], ThreaTrace [144]

the authors are used and assigned to the respective modules. In addition to the tabular classification, in Table 6, we discuss the individual characteristics of the building blocks, see Figure 2, based on the concrete manifestations of the systems.

*Data Collection.* Only systems that use the same basis are to be compared. Several have used CamFlow for monitoring so that a comparison between these systems could be meaningful. However, the underlying system and behavior must also be identical, which is difficult to achieve.

Since this problem of the data foundation is so prevalent, as shown in Table 5, it is noteworthy that systems with their own data collector component, on which the analysis is subsequently adapted, are tested with the available open data, which may have a significantly different structure.

This phenomenon becomes apparent when looking at the StreamSpot dataset [57], often used and created specifically for multi-graph recognition systems containing several traces. In contrast, other systems outperform this dataset's baseline using a single-host method. This is, however, expected due to the lack of better public comparisons of datasets or test environments.

The dataset problem also explains the frequent occurrence of custom, unfortunately mostly private, datasets with which authors evaluate their systems. Using private data allows for meaningful results as the data collection layer is reflected in the analysis results but often reduces reproducibility as the authors cannot publish the datasets for various reasons such as compliance or size.

Table 6. Integration into the Reference Architecture of the APT Attack Detection Systems

Paper	Data Collection	Data Storage and Management	Advanced Analytics	Reporting	Integration Layer	Extensibility and Adaptability Layer
StreamSpot [93]	-	Deletion by hyperparameter	Clustering-based anomaly detection	Substructures	-	-
NetWalk [164]	-	Fixed memory with reservoir sampling	Clustering-based uses NLP and random walks	Nodes	-	Continuous, since unsupervised
Log2vec [86]	System logs through rules into graph structure	-	Clustering-based uses NLP and random walks	Substructures	-	-
ThreaTrace [144]	CamFlow	GraphChi	Multi-GraphSAGE model	Nodes	Uses 2-hop neighborhood of anomalous nodes	Retain, add model, alert filtering
Pikachu [118]	-	Graphs of static snapshots	Autoencoder with RNN uses NLP, temporal walks	Edges	-	-
Li et al. [84]	Data parser	-	GNN autoencoder negative sampling trained	Edges	Hierarchical approach	-
Unicorn [58]	CamFlow	GraphChi and data retention	WL based, HistoSketch, clustering model	Time ranges	-	Retrain
FRAPpuccino [59]	CamFlow	GraphChi	Clustering through label-propagation	Nodes	-	-
Pagoda [155]	PASS	BerkeleyDB	Anomaly score based on rules with redis	Nodes	Data provision for forensic analyses	Add rules
ShadeWatcher [166]	Linux audit	Kafka with Postgres or Elasticsearch	Construct knowledge graph GNN recommender system	Edges	Visualization	User feedback
Sleuth [64]	Own graph collector for audit data	Own graph database	Tag and policy-based anomaly detection	Scenario graphs	Root-cause analysis for scenario graph building	Add policies
Provdetector [143]	Own collectors for Linux and Windows	Postgres	Density clustering with Doc2Vec embeddings	Substructures	-	Retrain
Kairos [30]	-	-	TGN with UniMP and rareness-scores	Scenario graphs	Correlate time windows based on anomalous and rare nodes	Incremental retraining

*Data Storage and Management.* Besides the data collection component, memory management functions and consumption are important, especially for longevity. There is a plurality of requirements from a detection system to successfully detect APT attacks, such as establishing links over a long window of time, apart from efficient collectors, who also require efficient storage that is capable of adapting since retaining all generated data over a long period is often impossible, especially in the case of long-running systems. The graph is an example of an efficient storage structure since the neighborhood of a node can be determined without considering time, unlike a time series, where time is decisive for the quantity to be considered. The graph is contextual enough to solve the backtracking needs of the algorithm so that main memory is manageable but is not necessarily more efficient in storage than storing data in other formats. For new events, the system must also *forget* unimportant events over

time, which will most likely never be associated with an attack. The classification shows that only a few systems offer solutions for this at all. The most sophisticated solution is Unicorn's, as prioritized forgetting is proposed.

*Analysis and Reporting.* For comparability of the analytical component, the conditions must allow it, like the same detection target. There is a big difference between a specific node, which, for instance, represents a file that is begin reported (ThreaTrace), or even small attack scenario graphs (Kairos), and a mere alarm is triggered (Unicorn). Thus, considering the attack, it is usually also not practical to achieve 100 % precision, which in most evaluations equates to identifying all events involved. In this sense, it is already difficult enough to identify the events in question manually in order to compare these values subsequently. Especially with the DARPA TC datasets, this problem becomes more evident, where very detailed provenance graph data are available but do not have labels. Only a text passage, the ground truth, in which the attack steps are explained, is available. From this point of view, it becomes clear that the actual goal is missed with the metrics used and that it is a historically developed comparison and evaluation methodology.

Regarding the metrics that evaluate the quality of detection, where commonly Precision, Accuracy, Recall, F1-Score, AUC, and ROC-Curve are referred to, and underlying input involving events, nodes, or log entries, depending on the system, they need to be reconsidered. A possible improvement is to make the evaluation more mission-oriented. If a detection system aims to detect APT attacks, we suggest paying closer attention to the requirements and the threat model. Thus, it is essential to link and detect inconspicuous and long-lived traces. It is desirable to report indicators that can expose the attacker as early as possible rather than determine all the events involved in the data exfiltration step at the end of the offensive campaign. Kairos tries to do this by not counting every event but dividing the data into time slices and counting the hits by reported time slice [30].

Thus, the quality of the recognition methods can only be evaluated in combination with the reporting, whereas the processing time and the considered information content, e.g., on the possible neighborhood to be considered, and so on, create comparability only on the component.

*Integration Layer.* In the case of the integration layer, there are currently few, if any, instances of this in the scientific prototypes. Here, integration is a useful method, as in the example of ThreaTrace, where the 2-hop neighborhood is considered [144], but is mainly helpful for adaptation to a production environment since this layer must integrate the unique APT attack detection with the target systems' specifics. The comparison only shows internal integrations, where the components interact. The main differences include additional data provision to improve the results' presentation or reuse of partial results. Approaches of such an integration can be seen with Kairos since here, in addition to detecting anomalies, a rareness factor is added, and the results are refined. All the indicators are then correlated and turned into scenario graphs for reporting [30]. Accordingly, proposed systems should only be evaluated for their ability to be integrated into live systems, which may not necessarily be a quality feature, and how effectively this layer is implemented.

*Extensibility and Adaptability Layer.* The same applies to the extensibility and adaptability layer, especially regarding structural extensions and adaptations. On the other hand, apparent differences in quality can be identified by adapting the analysis component. For example, adaptations possible at runtime, such as directly taking user feedback into account in the model, are helpful for an effective long-running detection system. The characteristics range from actual continuous extensions, as unsupervised clustering is used, to static management of allowlists and rule additions. Retraining is common, but ShadeWatcher also uses an adaptive learning approach [166].

## 8 Discussion

First, we would like to discuss the findings from the review on RQ1: *What are the specifics regarding the detection of an APT attack?*

We have described the threat landscape in detail in Section 3. The characteristics of APTs and their attacks are unique within the cyber threat landscape, with their decisiveness and *low and slow* modus operandi requiring

more nuanced approaches for successfully detecting and defending against them. Covert actions, such as insider threats or living-off-the-land tactics, should not be allowed to defeat detection mechanisms. In addition, aleatory uncertainty, such as zero-day exploits, must also be considered in some form, even if these are difficult to test in proposed systems. Due to the tendency of the attacker to act *low and slow*, correlations of benign-looking, individual inconsistencies, which, when compounded, reveal true APT activity, are regarded as particularly promising. In addition, integration with overarching continuous monitoring is necessary, and detection system creators should also consider being able to provide information about the system status on larger timescales common to APT attacks.

About RQ2: *How does a detection system address the particularities of an APT attack, and which existing approaches and systems are particularly suitable?*

The review revealed that no system for detecting APT attacks can yet fully detect and prevent all APT attacks. The results show that the systems capable of detecting APT attacks share a common architecture, as Section 6 illustrates. This architecture has been outlined as a reference architecture to establish better comparison metrics. Most of the systems we examined could not detect APT attacks in the sense of our definition because they made different, mainly weaker, assumptions about what an APT is. These systems must account for zero-day exploits, insider threats, and prolonged campaigns that can start at various points in the system and have diverse and often unpredictable goals. Detection methods are important, but a holistic understanding of security must be built by increasing situational awareness in all areas through defense in depth and breadth, as described in [22]. The most promising methods were those that detect deviations from normal behavior, since here no assumptions are made about the attackers, except that the attack differs from normal behavior.

Concerning the results of RQ3: *What criteria are used to evaluate the effectiveness of an APT attack detection system?*

We first discussed classifying the works in the review in the introductory evaluation in Section 5. We divided the evaluation into metrics, detection granularity, and datasets. We then examined the APT attack detection systems in more detail and looked specifically at the evaluation methods. We described the generally used metrics in Section 5.1. We noticed that metrics from standard ML evaluation, such as those calculated from the confusion matrix, were used. Occasionally, metrics were also used specifically for APT attacks, which were then detected campaigns [64]. The presented evaluation metrics Section 5.1 and the literature review used evaluation metrics shown in Table 5, which do not cover the requirements. For example, longevity is not considered, and the attack as a whole is neglected.

The review showed that datasets from DARPA were mainly used, and their self-defined datasets were not published. This shows that there is not yet a sufficient selection of reliable datasets for APT attack detection. The reviewed articles did not use the described methods of cyber range or an environment from which a dataset can be generated with configurations. The better the datasets are, the more targeted the evaluation of methods and the conclusions about their detection performance can be. A common conclusion that is particularly important is that the datasets can only be used for method evaluation and only provide starting points for how they work in the target domain. Field tests on the target domain can then evaluate the methods' suitability. The evaluation must be based on targeted metrics. The method should be adjusted to suit an evolving environment, similar to or matching the system to be protected.

The advantage of anomaly detection is that it can be tested in production without making scathing attacks on one's infrastructure. Each deviation from the regular operation should be reported. Thus, it is only necessary to check whether the normal behavior is mapped strongly enough or if other methods, which are not based on anomaly detection, are expressive enough. The first step for an improved evaluation is considering the proposed reference architecture to clarify the baseline situation.

To discuss RQ4: *What are the existing research limitations in APT attack detection, and where is the most potential?*

We have different scientific gaps in inadequate methods, insufficient consideration of the time factor, integration layers, and extensibility and adaptability layers.

In analyzing the detection methods used, those that rely on human knowledge stood out as particularly problematic, as attempting to combat APT attacks with CTI and similar methods appears counterintuitive because we are always one step behind the attacker. Surprisingly, we consider Sleuth and Pagoda viable APT attack detection systems. However, the literature review has shown that rule-based systems are rarely sufficient. It should be noted that rules and CTI are important in the integration layer of the reference architecture and can positively impact detection.

Additionally, some methods rely on supervised or semi-supervised learning and use examples to identify similar attacks. This may work well for conventional attacks but contradicts the characteristics of APT attacks. Another aspect is the selection of data sources. Traces and footprints of unauthorized or malicious activity are captured in many data sources. It is, therefore, counterintuitive to limit the scope of a dataset only to one type of data source, like network traffic. Instead, a holistic capture considers multiple and diverse sources and has intrinsically higher chances of capturing benign-looking interactions, which in fact are malicious.

Another aspect that emerges throughout the work is the consideration of temporal behavior. APT activity may span several months or even years. Thus, it is necessary to have a data storage and management component that, to some degree, captures it and methods that can create links over long periods. We found that this was where many solutions diverged. In the literature review, concepts such as approximated estimates [164], truncation of neighbors [93], or forgetting of presumably irrelevant nodes [58] were used sporadically.

Another point, also related to timing, is adaptability and extensibility. If an attack extends over a long time, the methods must also cover this period. Thus, changes to normal behavior must be considered if necessary. Methods ranging from manual sorting of results to allow and denylists, retraining, rule base expansion [64, 155], adding models [144], or dynamic updates through user feedback [166] were used in the review. The integration of users, considering user feedback, is rated by us as particularly good.

Given the growing volume of data, data reduction is crucial for practical applications. Various methods are used depending on the data source. Graphs as data structures help reduce raw data by avoiding multiple storage of entities. Tools like Drain [61] and graph-based reductions are useful for log data reduction and are also used in ShadeWatcher [166]. Network data is often grouped into flows as a reduction. Despite these strategies, managing the increasing data volume remains a challenge. Unicorn employs a forgetting strategy [58], which needs further examination in conjunction with data reduction.

Finally, we consider the integration layer. There is a lot of potential for optimization here, as the current work only considers its methods and, therefore, only creates integrations with internal modules. This layer includes, e.g., the enrichment of the alarm with the neighborhood of the suspicious node [144], graph data provision for backtracking of the alarm [64, 155], reuse of intermediate results [84], or the visualization of the results [166]. The approaches used here are also sporadically and represent only a selection of possibilities. External integrations have not been considered yet. This includes integrations with existing systems, such as a firewall or a connection to an SIEM system. The results can also be added to a knowledge graph to reduce false positives and provide additional information [76].

## 9 Future Directions

To enhance the practical feasibility of APT attack detection systems, future research should delve deeper into the individual components of the reference architecture, shown in Section 6. Regarding data collection, we expect future work to rely on graph data, similar to the studies examined here. Therefore, exploring data collection methods specifically tailored to graph data is crucial.

Data management becomes necessary in this context, as continuous operation and limited data storage capacity are important limitations of APT attack detection. Data reduction will be central to future data storage and management. Additionally, efficient data search capabilities are crucial for quick response times, particularly for graph-based methods requiring neighborhood information.

The untapped potential of methods for detecting anomalies should be explored further. So far, the main focus has been on self-supervised classifications, and with ShadeWatcher, an approach with negative sampling. Applying few-shot learning in this context is still uncharted territory. The inclusion of traditional methods, such as clustering, also still needs to be thoroughly investigated. Just like the combination of GNN with methods from other areas. Temporal data structures pose a particular challenge, as existing methods have not demonstrated consistent reliability. The need for new methods that can handle temporal data without the need for extensive evaluation or the addition of keywords is obvious. These methods should also be robust enough to adapt to changing environments. We are currently in the early stages of using anomaly detection to identify APT attacks.

The reporting of incidents must be further investigated in the context of graphs, with particular potential lying in visualization techniques. The authors of Kairos have already taken a substantial step in this direction. The integration of existing systems has not yet been thoroughly explored, and the extension and utilization of conventional detection systems also warrant further investigation. An important focus lies in the scalability and adaptability of the systems, as continuous runtime is essential when dealing with APT attacks.

In addition to the reference architecture components, the evaluation of these systems needs to be optimized for scientific comparison. While it is commendable that many systems provide their source code for re-evaluation, a more detailed description of the graph creation process, the utilized ground truth, and the establishment of standardized metrics is necessary. Furthermore, the currently employed datasets do not represent all possible attack scenarios or variations in normal behavior. Therefore, a stronger focus should be placed on utilizing testbeds and cyber ranges, enabling dynamic tests, and continuous operation testing.

## 10 Related Work

This section provides an overview of other studies, such as reviews and surveys, that are compared with our current efforts. In a comprehensive study, authors in [170] focus on **provenance-based IDSs (PIDSs)**. Apart from introducing a taxonomy of PIDS, they analyze the different aspects of provenance and graph-based analyses related to the subject in three core domain areas: data collection and processing methodologies, intrusion detection methodologies, and, finally, evaluation of PIDS and the corresponding benchmark datasets that various research teams have used. They conclude that despite the potential of PIDS in tackling APT-style attacks, they identify some issues for broader adoption of such methods, namely the overhead that such systems introduce to hosting platforms, the scalability and granularity of provenance collection systems, the challenges in developing reliable real-time detection engines, and the sore lack of benchmark datasets publicly available for researchers. Most research teams are forced to create their own evaluation datasets, which may be incomplete, falsely balanced, or otherwise problematic for such works. Even the DARPA TC datasets are usable up to an extent and are very difficult to work with.

Authors in [89] review some of the most recent PIDS. Similar to our work, they attempt to frame the structural components of PIDS systems in a unified design architecture. Our main differentiation point is the scope of APT attack detection, which we do not necessarily limit to PIDS and our understanding of the nature of the APT. We clearly distinguish between the threat, mostly from human actors, and the instantiation of the attacks, which can subsequently be detected in provenance data or other sources. We investigate methods to detect attacks that originate from APTs.

Similar work is performed by Zhenyuan Li et al. in [83], where provenance graph-based detection techniques are investigated. Detection techniques are divided into three general categories based on the detection approach: (i) tag propagation approaches, (ii) anomaly and outlier detection-based approaches, and (iii) graph matching-based approaches. The authors have a different view of the APT, so systems such as Holmes [100] are the focus of attention and are also classified as an APT detection system. However, Holmes does not focus on attack detection but only on the correlation of events for campaign detection. A comparison between systems like Holmes reveals, among others, the challenge of dependence explosion of provenance-based detection systems.

Simply put, dependence explosion is the phenomenon that occurs in provenance forward and backward tracking methods, where a long-running artifact, such as a process run, when causally related to malicious behavior, associates a large number of other artifacts, such as files and processes, with malicious activity, even though they may not necessarily be associated with it. The lack of benchmark datasets is also highlighted among the insights and challenges identified. Furthermore, the authors present a general framework for provenance-based threat detection systems, similar to our reference architecture. Ours differs because it is a universal reference architecture for APT attack detection systems, has two all-encompassing layers, and every log message goes through storage before the detection component is applied.

Regarding data provenance and its uses, Melanie Herschel et al. [63] survey its application fields in the published literature. The different types of provenance are presented, e.g., data and control provenance.

Detection of APT activity has also been attempted using graph generation and analysis techniques. In [151], the authors review the published literature surrounding the topic and expand on previously published surveys. Various approaches for attack graph generation and subsequent security analysis are visited. The general concept behind these approaches is that after vulnerabilities are identified in a system, the possible attack paths are consolidated in attack graphs. By correlating intrusion alerts to the graphs, APT attacks can be detected. As in the case of provenance graphs, the overhead induced and scalability of the proposed solutions are also challenging concerns. Similarly, Kerem Kaynar [69] presents a taxonomy of attack graph uses in network security.

In [110], the authors comprehensively review the domain of multi-step attack detection. Due to the character trait of APTs of operating in multiple steps, this work is highly relevant to our interests. Their findings suggest that most multi-step attack detection research is based on alert correlation. The lack of available public datasets is once again brought to the forefront of their conclusions, claiming that the majority of multi-step attack detection research evaluation is done using the DARPA 2000 dataset, which de facto is insufficient to capture our contemporary threat landscape concerning APTs. The authors also provide recommendations for improving the scientific integrity of future research in the domain, particularly regarding the reproducibility of experiments.

Adel Alshamrani et al. [6] survey the landscape of APTs by reviewing articles related to their nature, detection, and defense against them. Apart from presenting some case studies of APT attacks, they review various methods for attempting to detect APTs, including an extensive review of the role of ML in such efforts. Although their summary is not limited to graph-based detection techniques, they are featured specifically. The authors, however, see APTs as instances of an attack, which we generally do not accept, but rather as persons or groups of persons with malicious intent and advanced capabilities. A clear separation must emerge between the (AP)threat and the actual attack. Threats, which refer to groups, people, and others, are considered when planning cyber defense strategies. Detection is specifically about concrete attacks of APTs. Though they argue that definitions of APTs are relatively narrow, we also find them valid, as seen by the broader scope of the NIST cyber-resiliency framework [15] definition. We are particularly critical of the definition of APTs since the authors also describe what an APT is not, and only technically very sophisticated attacks are considered APTs. However, we see the adaptation to the target as a special feature, where an APT can cause serious damage even with simple means.

In [36], the authors conceived a development roadmap for APT attack detection. In 2012, the authors identified the APT threat landscape and the necessary steps many works dedicated to APT attack detection fail to consider. The authors describe that different data sources and analysis methods are sometimes better suited than others for the different steps of an APT attack. They also emphasize the added value of combining methods and data sources. In general, our review confirms the core statements of the 2012 framework, addressing specific characteristics and considering novel methods. However, similar emphases were made in 2012 and still need to be implemented in current work.

Our findings align with a recent survey by Bilot et al. [17], emphasizing graphs and GNN potential in attack detection. Their comprehensive survey provides a more detailed exploration of various methods that can be utilized for this task, surpassing the level of detail we covered. While our focus was primarily on APT attack detection and our deduction of GNN promise in this context, their survey encompasses attack detection in a

broader sense, with a different focus. They also classify detection into network and host detection, considering hybrid approaches as a special case. For APT attack detection, we believe that hybrid methods are essential, as attackers can exploit various forms of communication based on the characteristics of APTs. The authors similarly conclude that GNNs hold great promise.

Lastly, Khalid et al. [73] summarize some attributes of APT attacks and present an overview of rule-based versus anomaly-based detection schemes. While their presentation of state-of-the-art APT detection is relatively shallow, the challenges identified hold merit. They focus on the nature of the APT and the attributes that hinder effective detection, especially for unknown attack vectors.

## 11 Conclusion

This article presents a review of the topic of APT attack detection systems. We created a literature corpus of 70 papers using the structured literature search with a forward and backward citation search. We performed a systematization along the threat model, detection methodology, and data type dimensions.

The threat model was most important since the detection should relate to APT attacks. We distinguished APT attacks from multi-step and conventional attacks, addressed their sophistication, longevity, and determination characteristics, and common methods were discussed and put into the context of APT attack detection. Linking data points that are distant in time and looking at data points from different sources considered together offer great potential to detect sophisticated, long-lived, distributed, covert attacks. DL anomaly detection methods, especially GNN, show the greatest potential. Considering datasets, the current challenges and shortcomings were discussed by presenting the existing datasets for APT attack detection system evaluation. Currently, few are meaningful and provide a solid ground for comparison, as a result of the fact that DARPA's TC program datasets are particularly popular in current research.

This review contributes to the scientific community by providing a comprehensive overview of existing systems, concepts, and methods related to APT attack detection. The research covers essential aspects, including the challenges associated with APT attack detection, the underlying detection mechanisms, and the evaluation concepts encompassing various granularities, metrics, and datasets. Moreover, the review explains APT attack detection systems and the scientific gaps that previous work has revealed. We generated a reference architecture as an additional outcome of this literature review. Together, these findings aim to assist researchers in this field with a comprehensive understanding, enabling them to develop new and improved systems for APT attack detection.

Revisiting our initial RQs, we draw the following conclusions: No system for detecting APT attacks is operational. The peculiarities of APT attacks make it necessary to revise the prevailing methods for their detection and adapt them, especially for their long runtime. In addition, a unified evaluation is needed. Lastly, the fluidity of interpretations of what the APT is and does is high, leading to substantial diversity in the detection objectives of proposed systems, which further encumbers direct comparison. This needs to be unified; our definition of an APT and the distinction to an APT attack are outlined in the article.

## References

- [1] Asad Afreen, Moosa Aslam, and Saad Ahmed. 2020. Analysis of fileless malware and its evasive behavior. In *Proceedings of the International Conference on Cyber Warfare and Security (ICWS '20)*. IEEE, 1–8. DOI: <https://doi.org/10.1109/ICWS48432.2020.9292376>
- [2] Olusola Akinrolabu, Ioannis Agrafiotis, and Arnau Erola. 2018. The challenge of detecting sophisticated attacks: Insights from SOC ANALYSTS. In *Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES '18)*. ACM, New York, NY, Article 55, 9 pages. DOI: <https://doi.org/10.1145/3230833.3233280>
- [3] Sumayah Al-Rabiaah. 2018. The “Stuxnet” virus of 2010 as an example of a “APT” and its “Recent” variances. In *Proceedings of the 21st Saudi Computer Society National Computer Conference (NCC '18)*. IEEE, 1–5. DOI: <https://doi.org/10.1109/NCG.2018.8593143>
- [4] Almuthanna Alageel and Sergio Maffei. 2021. Hawk-eye: Holistic detection of APT command and control domains. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC '21)*. ACM, New York, NY, 1664–1673. DOI: <https://doi.org/10.1145/3412841.3442040>

- [5] Abdulleh Alsaaheel, Yuhong Nan, Shiqing Ma, Le Yu, Gregory Walkup, Z. Berkay Celik, Xiangyu Zhang, and Dongyan Xu. 2021. ATLAS: A sequence-based learning approach for attack investigation. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21)*. USENIX Association, 3005–3022. Retrieved from <https://www.usenix.org/conference/usenixsecurity21/presentation/alsaaheel>
- [6] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. 2019. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials* 21, 2 (2019), 1851–1877. DOI: <https://doi.org/10.1109/COMST.2019.2891891>
- [7] Md. Monowar Anjum, Shahrear Iqbal, and Benoit Hamelin. 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 (SACMAT '21)*. ACM, New York, NY, 27–32. DOI: <https://doi.org/10.1145/3450569.3463573>
- [8] Giovanni Apruzzese, Pavel Laskov, Edgardo Montes de Oca, Wissam Mallouli, Luis Brdalo Rapa, Athanasios Vasileios Grammatopoulos, and Fabio Di Franco. 2023. The role of machine learning in cybersecurity. *Digital Threats* 4, 1 (Mar. 2023), Article 8, 38 pages. DOI: <https://doi.org/10.1145/3545574>
- [9] Rody Arantes, Carl Weir, Henry Hannon, and Marisha Kulseng. 2021. Operationally transparent cyber (OpTC). *IEEE Dataport*. DOI: <https://doi.org/10.21227/edq8-nk52>
- [10] Emmanuel Aroms. 2012. *NIST Special Publication 800-39 Managing Information Security Risk*. CreateSpace, Scotts Valley, CA. Retrieved from <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-39.pdf>
- [11] M. Ashwini Kumari and K. S. Nandini Prasad. 2021. A behavioral study of advanced security attacks in enterprise networks. In *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. IEEE, Bangalore, India, 1–5. DOI: <https://doi.org/10.1109/CSITSS54238.2021.9682903>
- [12] Urooj Aslam, Ezzat Batool, Syed Ahsan, and Abdullah Sultan. 2017. Hybrid network intrusion detection system using machine learning classification and rule based learning system. *International Journal of Grid and Distributed Computing* 10 (Feb. 2017), 51–62. DOI: <https://doi.org/10.14257/ijgcd.2017.10.2.05>
- [13] Gbadebo Ayoade, Khandakar Ashrafi Akbar, Pracheta Sahoo, Yang Gao, Anmol Agarwal, Kangkook Jee, Latifur Khan, and Anoop Singhal. 2020. Evolving advanced persistent threat detection using provenance graph and metric learning. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS '20)*. IEEE, 1–9. DOI: <https://doi.org/10.1109/CNS48642.2020.9162264>
- [14] Mathieu Barre, Ashish Gehani, and Vinod Yegneswaran. 2019. Mining data provenance to detect advanced persistent threats. In *Proceedings of the 11th USENIX Conference on Theory and Practice of Provenance (TAPP '19)*. USENIX Association, 6. Retrieved from <https://www.usenix.org/conference/tapp2019/presentation/barre>
- [15] Matthew Barrett. 2018. Framework for improving critical infrastructure cybersecurity version 1.1. *National Institute of Standards and Technology*. DOI: <https://doi.org/10.6028/NIST.CSWP.04162018>
- [16] Richard Bejtlich. 2010. What APT Is. Retrieved from [https://www.academia.edu/6842130/What\\_APT\\_Is](https://www.academia.edu/6842130/What_APT_Is)
- [17] Tristan Bitol, Nour El Madhoun, Khaldoun Al Agha, and Anis Zouaoui. 2023. Graph neural networks for intrusion detection: A survey. *IEEE Access* 11 (2023), 49114–49139. DOI: <https://doi.org/10.1109/ACCESS.2023.3275789>
- [18] Benjamin Bowman and H. Howie Huang. 2021. Towards next-generation cybersecurity with graph AI. *SIGOPS Operating Systems Review* 55, 1 (Jun. 2021), 61–67. DOI: <https://doi.org/10.1145/3469379.3469386>
- [19] Benjamin Bowman, Craig Laprade, Yuede Ji, and H. Howie Huang. 2020. Detecting lateral movement in enterprise computer networks with unsupervised graph AI. In *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID '20)*. USENIX Association, 257–268. Retrieved from <https://www.usenix.org/conference/raid2020/presentation/bowman>
- [20] Kenneth Brancik and Gabriel Ghinita. 2011. The optimization of situational awareness for insider threat detection. In *Proceedings of the 1st ACM Conference on Data and Application Security and Privacy (CODASPY '11)*. ACM, New York, NY, 231–236. DOI: <https://doi.org/10.1145/1943513.1943544>
- [21] Guillaume Brogi and Valerie Viet Triem Tong. 2016. TerminAPTor: Highlighting advanced persistent threats through information flow tracking. In *Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS '16)*. IEEE, 1–5. DOI: <https://doi.org/10.1109/NTMS.2016.7792480>
- [22] Robin Buchta, Felix Heine, and Carsten Kleiner. 2022. Challenges and peculiarities of attack detection in virtual power plants: Towards an advanced persistent threat detection system. In *Proceedings of the IEEE 29th Annual Software Technology Conference (STC '22)*. IEEE, 69–81. DOI: <https://doi.org/10.1109/STC55697.2022.00019>
- [23] Sergio Caltagirone, Andrew D. Pendergast, and Chris Betz. 2013. The Diamond Model of Intrusion Analysis. Retrieved from <https://apps.dtic.mil/sti/pdfs/ADA586960.pdf>
- [24] Hasan Cam. 2020. Cyber resilience using autonomous agents and reinforcement learning. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*. Tien Pham, Latasha Solomon, and Katie Rainey (Eds.), Vol. 11413. International Society for Optics and Photonics, SPIE, Online Only, 114130R. DOI: <https://doi.org/10.1117/12.2559319>
- [25] A. A. Cardenas, J. S. Baras, and K. Seamon. 2006. A framework for the evaluation of intrusion detection systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S & P '06)*. IEEE, 15. DOI: <https://doi.org/10.1109/SP.2006.2>

- [26] Defense Use Case. 2016. Analysis of the cyber attack on the Ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)* 388 (2016), 1–29. Retrieved from <https://nsarchive.gwu.edu/sites/default/files/documents/3891751/SANS-and-Electricity-Information-Sharing-and.pdf>
- [27] Microsoft Security Response Center. 2020. Corporate IoT - A Path to Intrusion. Retrieved November 14, 2023 from <https://msrc-blog.microsoft.com/2019/08/05/corporate-iot-a-path-to-intrusion/>
- [28] Saranya Chandran, P. Hrudya, and Prabaharan Poornachandran. 2015. An efficient classification model for detecting advanced persistent threat. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '15)*. IEEE, 2001–2009. DOI: <https://doi.org/10.1109/ICACCI.2015.7275911>
- [29] Yen-Yu Chang, Pan Li, Rok Sosic, M. H. Afifi, Marco Schweighauser, and Jure Leskovec. 2021. F-FADE: Frequency factorization for anomaly detection in edge streams. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. ACM, New York, NY, 589–597. DOI: <https://doi.org/10.1145/3437963.3441806>
- [30] Zijun Cheng, Qiujuan Lv, Jinyuan Liang, Yan Wang, Degang Sun, Thomas Pasquier, and Xueyuan Han. 2024. Kairos: Practical intrusion detection and investigation using whole-system provenance. In *Proceedings of the IEEE Symposium on Security and Privacy (S & P '24)*. IEEE, 9–9. DOI: <https://doi.org/10.1109/TIFS.2022.3208815>
- [31] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. DOI: <https://doi.org/10.3115/v1/D14-1179>
- [32] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. Retrieved from [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf)
- [33] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyszogrod, and M. A. Zissman. 1999. Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation. Retrieved from <https://apps.dtic.mil/sti/pdfs/ADA526274.pdf>
- [34] Cybersecurity and Infrastructure Security Agency (CISA). 2022. APT Cyber Tools Targeting ICS/SCADA Devices. Retrieved November 14, 2023 from <https://www.cisa.gov/uscert/ncas/alerts/aa22-103a>
- [35] darpa-i2o. 2020. Transparent-Computing: Material from the DARPA Transparent Computing Program. Retrieved November 14, 2023 from <https://github.com/darpa-i2o/Transparent-Computing>
- [36] Johannes de Vries, Hans Hoogstraaten, Jan van den Berg, and Semir Daskapan. 2012. Systems for detecting advanced persistent threats: A development roadmap using intelligent data analysis. In *Proceedings of the 2012 International Conference on Cyber Security*. IEEE, 54–61. DOI: <https://doi.org/10.1109/CyberSecurity.2012.14>
- [37] Thibault Debatty, Wim Mees, and Thomas Sison. 2018. Graph-based APT detection. In *Proceedings of the International Conference on Military Communications and Information Systems (ICMCIS '18)*. IEEE, 1–8. DOI: <https://doi.org/10.1109/ICMCIS.2018.8398708>
- [38] R. Deibert, R. Rohozinski, A. Manchanda, N. Villeneuve, and G. Walton. 2009. Tracking GhostNet: Investigating a cyber espionage network. Retrieved from <https://ora.ox.ac.uk/objects/uuid:6d1260fd-b8ee-4a11-8a5f-e7708d543651/files/m0e5334ee23f2b6390f1a69c5d791e618>
- [39] Alessandro Di Pinto, Younes Dragoni, and Andrea Carcano. 2018. TRITON: The first ICS cyber attack on safety instrument systems. In *Proceedings of the Black Hat USA*, 26 pages. Retrieved from <https://i.blackhat.com/us-18/Wed-August-8/us-18-Carcano-TRITON-How-It-Disrupted-Safety-Systems-And-Changed-The-Threat-Landscape-Of-Industrial-Control-Systems-Forever-wp.pdf>
- [40] Dragos Inc. 2022. Pipedream: Chernovite’s Emerging Malware Targeting Industrial Control Systems. Retrieved from [https://hub.dragos.com/hubfs/116-Whitepapers/Dragos\\_ChernoviteWP\\_v2b.pdf?hsLang=en](https://hub.dragos.com/hubfs/116-Whitepapers/Dragos_ChernoviteWP_v2b.pdf?hsLang=en)
- [41] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, New York, NY, 1285–1298. DOI: <https://doi.org/10.1145/3133956.3134015>
- [42] Hope Eke, Andrei Petrovski, and Hatem Ahriz. 2020. Detection of false command and response injection attacks for cyber physical systems security and resilience. In *Proceedings of the 13th International Conference on Security of Information and Networks (SIN '20)*. ACM, New York, NY, Article 10, 8 pages. DOI: <https://doi.org/10.1145/3433174.3433615>
- [43] Hope Nkiruka Eke, Andrei Petrovski, and Hatem Ahriz. 2019. The use of machine learning algorithms for detecting advanced persistent threats. In *Proceedings of the 12th International Conference on Security of Information and Networks (SIN '19)*. ACM, New York, NY, Article 5, 8 pages. DOI: <https://doi.org/10.1145/3357613.3357618>
- [44] Dhivya Eswaran and Christos Faloutsos. 2018. SedanSpot: Detecting anomalies in edge streams. In *Proceedings of the IEEE International Conference on Data Mining (ICDM '18)*. IEEE, 953–958. DOI: <https://doi.org/10.1109/ICDM.2018.00117>
- [45] Gilberto Fernandes, Joel J. P. C. Rodrigues, Luiz Fernando Carvalho, Jalal F. Al-Muhtadi, and Mario Lemes Proença. 2018. A comprehensive survey on network anomaly detection. *Telecommunication Systems* 70, 3 (Jul. 2018), 447–489. DOI: <https://doi.org/10.1007/s11235-018-0475-8>

- [46] Zarestel Ferrer and Methusela Cebrian Ferrer. 2010. In-depth Analysis of Hydraq. Retrieved from [https://paper.seebug.org/papers/APT/APT\\_CyberCriminal\\_Campagin/2010/in-depth\\_analysis\\_of\\_hydraq\\_final\\_231538.pdf](https://paper.seebug.org/papers/APT/APT_CyberCriminal_Campagin/2010/in-depth_analysis_of_hydraq_final_231538.pdf)
- [47] FiveDirections. 2020. OpTC-Data. Retrieved November 14, 2023 from <https://github.com/FiveDirections/OpTC-data>
- [48] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J. Aparicio-Navarro. 2018. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems* 89 (2018), 349–359. DOI : <https://doi.org/10.1016/j.future.2018.06.055>
- [49] Ibrahim Ghafir, Konstantinos G. Kyriakopoulos, Sangarapillai Lambotharan, Francisco J. Aparicio-Navarro, Basil Assadhan, Hamad Binsalleeh, and Diab M. Diab. 2019. Hidden Markov models and alert correlations for the prediction of advanced persistent threats. *IEEE Access* 7 (2019), 99508–99520. DOI : <https://doi.org/10.1109/ACCESS.2019.2930200>
- [50] Paul Giura and Wei Wang. 2012. A context-based detection framework for advanced persistent threats. In *Proceedings of the International Conference on Cyber Security*. IEEE, 69–74. DOI : <https://doi.org/10.1109/CyberSecurity.2012.16>
- [51] Joshua Glasser and Brian Lindauer. 2013. Bridging the gap: A pragmatic approach to generating insider threat data. In *Proceedings of the IEEE Security and Privacy Workshops*. IEEE, 98–104. DOI : <https://doi.org/10.1109/SPW.2013.37>
- [52] John Griffith, Derrick Kong, Armando Caro, Brett Benyo, Joud Khoury, Timothy Upthegrove, Timothy Christovich, Stanislav Ponomorov, Ali Sydney, Arjun Saini, Vladimir Shurbanov, Christopher Willig, David Levin, and Jack Dietz. 2020. Scalable Transparency Architecture for Research Collaboration (STARC) – DARPA Transparent Computing (TC) Program. Retrieved from <https://apps.dtic.mil/sti/pdfs/AD1092961.pdf>
- [53] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, 855–864. DOI : <https://doi.org/10.1145/2939672.2939754>
- [54] Juan Andres Guerrero-Saade, Costin Raiu, Daniel Moore, and Thomas Rid. 2017. PENQUIN’S MOONLIT MAZE The Dawn of Nation-State Digital Espionage. Retrieved November 14, 2023 from [https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07180251/Penquins\\_Moonlit\\_Maze\\_PDF\\_eng.pdf](https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07180251/Penquins_Moonlit_Maze_PDF_eng.pdf)
- [55] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the Advances in Neural Information Processing Systems*. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., Retrieved from [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9bea9-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9bea9-Paper.pdf)
- [56] Dongqi Han, Zhiliang Wang, Wenqi Chen, Ying Zhong, Su Wang, Han Zhang, Jiahai Yang, Xingang Shi, and Xia Yin. 2021. DeepAID: Interpreting and improving deep learning-based anomaly detection in security applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*. ACM, New York, NY, 3197–3217. DOI : <https://doi.org/10.1145/3460120.3484589>
- [57] Xueyuan Han. 2018. StreamSpot dataset. *Harvard Dataverse*. DOI : <https://doi.org/10.7910/DVN/83KYJY>
- [58] Xueyuan Han, Thomas Pasquier, Adam Bates, James Mickens, and Margo Seltzer. 2020. Unicorn: Runtime provenance-based detector for advanced persistent threats. In *Proceedings 2020 Network and Distributed System Security Symposium*. Internet Society, 18. DOI : <https://doi.org/10.14722/ndss.2020.24046>
- [59] Xueyuan Han, Thomas Pasquier, Tanvi Ranjan, Mark Goldstein, and Margo Seltzer. 2017. FRAppuccino: Fault-detection through runtime analysis of provenance. In *Proceedings of the 9th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '17)*. USENIX Association, 7. Retrieved from <https://www.usenix.org/conference/hotcloud17/program/presentation/han>
- [60] Wajih Ul Hassan, Shengjian Guo, Ding Li, Zhengzhang Chen, Kangkook Jee, Zhichun Li, and Adam Bates. 2019. NoDoze: Combatting threat alert fatigue with automated provenance triage. In *Proceedings 2019 Network and Distributed System Security Symposium*. The Internet Society, 7. DOI : <https://doi.org/10.14722/ndss.2019.23349>
- [61] Pinjia He, Jieming Zhu, Zibin Zheng, and Michael R. Lyu. 2017. Drain: An online log parsing approach with fixed depth tree. In *Proceedings of the IEEE International Conference on Web Services (ICWS '17)*. IEEE, 33–40. DOI : <https://doi.org/10.1109/ICWS.2017.13>
- [62] Felix Heine, Tim Laue, and Carsten Kleiner. 2020. On the evaluation and deployment of machine learning approaches for intrusion detection. In *Proceedings of the IEEE International Conference on Big Data (Big Data '20)*. IEEE, 4594–4603. DOI : <https://doi.org/10.1109/BigData50022.2020.9378479>
- [63] Melanie Herschel, Ralf Diestelkämper, and Housseem Ben Lahmar. 2017. A survey on provenance: What for? What form? What from? *The VLDB Journal* 26, 6 (Dec. 2017), 881–906. DOI : <https://doi.org/10.1007/s00778-017-0486-1>
- [64] Md Nahid Hossain, Sadegh M. Milajerdi, Junao Wang, Birhanu Eshete, Rigel Gjomemo, R. Sekar, Scott Stoller, and V. N. Venkatakrishnan. 2017. SLEUTH: Real-time attack scenario reconstruction from COTS audit data. In *Proceedings of the 26th USENIX Security Symposium (USENIX Security '17)*. USENIX Association, 487–504. Retrieved from <https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/hossain>
- [65] Wenpeng Hu, Mengyu Wang, Qi Qin, Jinwen Ma, and Bing Liu. 2020. HRN: A holistic approach to one class learning. In *Proceedings of the Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 19111–19124. Retrieved from <https://proceedings.neurips.cc/paper/2020/file/dd1970fb03877a235d530476eb727dab-Paper.pdf>

- [66] Piotr Indyk and Rameez Motwani. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC '98)*. ACM, New York, NY, 604–613. DOI : <https://doi.org/10.1145/276698.276876>
- [67] Hassaan Irshad, Gabriela Ciocarlie, Ashish Gehani, Vinod Yegneswaran, Kyu Hyung Lee, Jignesh Patel, Somesh Jha, Yonghwi Kwon, Dongyan Xu, and Xiangyu Zhang. 2021. TRACE: Enterprise-wide provenance tracking for real-time APT detection. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4363–4376. DOI : <https://doi.org/10.1109/TIFS.2021.3098977>
- [68] Jianguo Jiang, Jiuming Chen, Tianbo Gu, Kim-Kwang Raymond Choo, Chao Liu, Min Yu, Weiqing Huang, and Prasant Mohapatra. 2019. Anomaly detection with graph convolutional networks for insider threat and fraud detection. In *Proceedings of the IEEE Military Communications Conference (MILCOM '19)*. IEEE, 109–114. DOI : <https://doi.org/10.1109/MILCOM47813.2019.9020760>
- [69] Kerem Kaynar. 2016. A taxonomy for attack graph generation and usage in network security. *Journal of Information Security and Applications* 29 (2016), 27–56. DOI : <https://doi.org/10.1016/j.jisa.2016.02.001>
- [70] Alexander D. Kent. 2015. *Comprehensive, Multi-Source Cyber-Security Events*. Los Alamos National Laboratory. DOI : <https://doi.org/10.17021/1179829>
- [71] Alexander D. Kent. 2016. *Cyber Security Data Sources for Dynamic Network Research*. World Scientific Europe, Chapter Chapter 2, 37–65. DOI : [https://doi.org/10.1142/9781786340757\\_0002](https://doi.org/10.1142/9781786340757_0002)
- [72] Adam Khalid, Anazida Zainal, Mohd Aizaini Maarof, and Fuad A. Ghaleb. 2021. Advanced persistent threat detection: A survey. In *Proceedings of the 3rd International Cyber Resilience Conference (CRC '21)*. IEEE, 1–6. DOI : <https://doi.org/10.1109/CRC50527.2021.9392626>
- [73] Joud Khoury, Timothy Upthegrove, Armando Caro, Brett Benyo, and Derrick Kong. 2020. An event-based data model for granular information flow tracking. In *Proceedings of the 12th International Workshop on Theory and Practice of Provenance (TaPP '20)*. USENIX Association, Online Only, 6. Retrieved from <https://www.usenix.org/conference/tapp2020/presentation/khoury>
- [74] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* 2, 1 (Jul. 2019), 20. DOI : <https://doi.org/10.1186/s42400-019-0038-7>
- [75] Samuel T. King and Peter M. Chen. 2003. Backtracking Intrusions. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP '03)*. ACM, New York, NY, 223–236. DOI : <https://doi.org/10.1145/945445.945467>
- [76] Kabul Kurniawan, Andreas Ekelhart, Elmar Kiesling, Gerald Quirchmayr, and A. Min Tjoa. 2022. KRYSTAL: Knowledge graph-based framework for tactical attack discovery in audit data. *Computers & Security* 121, C (Oct. 2022), 19 pages. DOI : <https://doi.org/10.1016/j.cose.2022.102828>
- [77] Aapo Kyrola, Guy Blelloch, and Carlos Guestrin. 2012. GraphChi: Large-scale graph computation on just a PC. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation (OSDI '12)*. USENIX Association, 31–46. Retrieved from <https://www.usenix.org/system/files/conference/osdi12/osdi12-final-126.pdf>
- [78] Giuseppe Laurenza, Riccardo Lazzeretti, and Luca Mazzotti. 2020. Malware triage for early identification of advanced persistent threat activities. *Digital Threats* 1, 3 (Aug. 2020), Article 16, 17 pages. DOI : <https://doi.org/10.1145/3386581>
- [79] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32. Eric P. Xing and Tony Jebara (Eds.), PMLR, 1188–1196. Retrieved from <https://proceedings.mlr.press/v32/le14.html>
- [80] John Boaz Lee, Giang Nguyen, Ryan A. Rossi, Nesreen K. Ahmed, Eunye Koh, and Sungchul Kim. 2021. Dynamic node embeddings from edge streams. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 6 (2021), 931–946. DOI : <https://doi.org/10.1109/TETCI.2020.3011432>
- [81] Jong-Hoon Lee, Ik Kyun Kim, and Ki-Jun Han. 2015. An abnormal connection detection system based on network flow analysis. In *Proceedings of the IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin '15)*. IEEE, 71–75. DOI : <https://doi.org/10.1109/ICCE-Berlin.2015.7391336>
- [82] Yi-Fan Li, Yang Gao, Gbadebo Ayoade, Hemeng Tao, Latifur Khan, and Bhavani Thuraisingham. 2019. Multistream classification for cyber threat data with heterogeneous feature space. In *Proceedings of the World Wide Web Conference (WWW '19)*. ACM, New York, NY, 2992–2998. DOI : <https://doi.org/10.1145/3308558.3313572>
- [83] Zhenyuan Li, Qi Alfred Chen, Runqing Yang, Yan Chen, and Wei Ruan. 2021. Threat detection and investigation with system-level provenance graphs: A survey. *Computers & Security* 106, C (Jul. 2021), 16 pages. DOI : <https://doi.org/10.1016/j.cose.2021.102282>
- [84] Zitong Li, Xiang Cheng, Lixiao Sun, Ji Zhang, Bing Chen, and Weizhi Meng. 2021. A hierarchical approach for advanced persistent threat detection with attention-based graph neural networks. *Security and Communication Networks* 2021 (Jan. 2021), 14 pages. DOI : <https://doi.org/10.1155/2021/9961342>
- [85] Brian Lindauer. 2020. Insider Threat Test Dataset. Retrieved November 14, 2023 from [https://kithub.cmu.edu/articles/dataset/Insider\\_Threat\\_Test\\_Dataset/12841247/1](https://kithub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1)
- [86] Fucheng Liu, Yu Wen, Dongxue Zhang, Xihe Jiang, Xinyu Xing, and Dan Meng. 2019. Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. ACM, New York, NY, 1777–1794. DOI : <https://doi.org/10.1145/3319535.3363224>

- [87] Hongyu Liu and Bo Lang. 2019. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences* 9, 20 (2019), 28. DOI : <https://doi.org/10.3390/app9204396>
- [88] Robert Luh, Sebastian Schrittwieser, and Stefan Marschalek. 2016. TAON: An ontology-based approach to mitigating targeted attacks. In *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services (iiWAS '16)*. ACM, New York, NY, 303–312. DOI : <https://doi.org/10.1145/3011141.3011157>
- [89] Yang Lv, Shaona Qin, Zifeng Zhu, Zhuocheng Yu, Shudong Li, and Weihong Han. 2022. A review of provenance graph based APT attack detection: Applications and developments. In *Proceedings of the 7th IEEE International Conference on Data Science in Cyberspace (DSC '22)*. IEEE, 498–505. DOI : <https://doi.org/10.1109/DSC55868.2022.00075>
- [90] Zhen Ma, Qiang Li, and Xiangyu Meng. 2019. Discovering suspicious APT families through a large-scale domain graph in information-centric IoT. *IEEE Access* 7 (2019), 13917–13926. DOI : <https://doi.org/10.1109/ACCESS.2019.2894509>
- [91] Aine MacDermott, Phillip Kendrick, Ibrahim Idowu, Mal Ashall, and Qi Shi. 2019. Securing things in the healthcare internet of things. In *Proceedings of the Global IoT Summit (GloTS '19)*. IEEE, 1–6. DOI : <https://doi.org/10.1109/GIOTS.2019.8766383>
- [92] Muhammad Salahuddin Manggalanny and Kalamullah Ramli. 2017. Combination of DNS traffic analysis: A design to enhance APT detection. In *Proceedings of the 3rd International Conference on Science and Technology - Computer (ICST '17)*. IEEE, 171–175. DOI : <https://doi.org/10.1109/ICSTC.2017.8011873>
- [93] Emaad Manzoor, Sadeq M. Milajerdi, and Leman Akoglu. 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 (KDD '16)*. ACM, New York, NY, 1035–1044. DOI : <https://doi.org/10.1145/2939672.2939783>
- [94] Lockheed Martin. 2023. The Cyber Kill Chain. Retrieved from <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
- [95] Lennart Maschmeyer and Myriam Dunn Cavelt. 2022-05. Goodbye cyberwar: Ukraine as reality check. *CSS Policy Perspectives* 10, 3. DOI : <https://doi.org/10.3929/ethz-b-000549252>
- [96] Wim Mees and Thibault Debatty. 2014. Multi-agent system for APT detection. In *Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops*. IEEE, 401–406. DOI : <https://doi.org/10.1109/ISSREW.2014.86>
- [97] Peter Mell, R. Lippmann, Chung Tong Hu, J. Haines, and M. Zissman. 2003. An overview of issues in testing intrusion detection systems. In *NIST Interagency/Internal Report (NISTIR)*. National Institute of Standards and Technology, Gaithersburg, MD. DOI : <https://doi.org/10.6028/NIST.IR.7007>
- [98] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781 [cs.CL]. Retrieved from <https://doi.org/10.48550/arXiv.1301.3781>
- [99] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS '13)*, Vol. 2. Curran Associates Inc., 3111–3119. DOI : <https://doi.org/10.1145/1943513.1943544>
- [100] Sadeq M. Milajerdi, Rigel Gjomemo, Birhanu Eshete, R. Sekar, and V. N. Venkatakrishnan. 2019. HOLMES: Real-time APT detection through correlation of suspicious information flows. In *Proceedings of the IEEE Symposium on Security and Privacy (SP '19)*. IEEE, 1137–1152. DOI : <https://doi.org/10.1109/SP.2019.00026>
- [101] Preeti Mishra, Vijay Varadharajan, Uday Tupakula, and Emmanuel S. Pilli. 2019. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys & Tutorials* 21, 1 (2019), 686–728. DOI : <https://doi.org/10.1109/COMST.2018.2847722>
- [102] MITRE. 2022. MITRE Cyber Analytics Repository. Retrieved November 14, 2023 from <https://car.mitre.org>
- [103] Nachaat Mohamed and Bahari Belaton. 2021. SBI model for the detection of advanced persistent threat based on strange behavior of using credential dumping technique. *IEEE Access* 9 (2021), 42919–42932. DOI : <https://doi.org/10.1109/ACCESS.2021.3066289>
- [104] Ned Moran. 2010. Understanding advanced persistent threats: A case study. *login* 36, 4 (2010), 21–26. Retrieved from <https://www.usenix.org/system/files/login/articles/105484-Moran.pdf>
- [105] Nour Moustafa and Jill Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *Proceedings of the Military Communications and Information Systems Conference (MilCIS '15)*. IEEE, 1–6. DOI : <https://doi.org/10.1109/MilCIS.2015.7348942>
- [106] Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Margo Seltzer. 2006. Provenance-aware storage systems. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference (ATEC '06)*. USENIX Association, 4. DOI : <https://doi.org/10.5555/1267359.1267363>
- [107] Sowmya Myneni, Ankur Chowdhary, Abdulhakim Sabur, Sailik Sengupta, Garima Agrawal, Dijiang Huang, and Myong Kang. 2020. DAPT 2020 - Constructing a benchmark dataset for advanced persistent threats. In *Deployable Machine Learning for Security Defense*. Springer International Publishing, 138–163. DOI : [https://doi.org/10.1007/978-3-030-59621-7\\_8](https://doi.org/10.1007/978-3-030-59621-7_8)
- [108] N. Thomas Rincy and Roopam Gupta. 2020. A survey on machine learning approaches and its techniques. In *Proceedings of the IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCECS '20)*. IEEE, 1–6. DOI : <https://doi.org/10.1109/SCECS48394.2020.190>

- [109] Nitin Naik, Paul Jenkins, Paul Grace, and Jingping Song. 2022. Comparing attack models for IT systems: Lockheed Martin’s cyber kill chain, MITRE ATT & CK framework and diamond model. In *Proceedings of the IEEE International Symposium on Systems Engineering (ISSE ’22)*. IEEE, 1–7. DOI: <https://doi.org/10.1109/ISSE54508.2022.10005490>
- [110] Julio Navarro, Aline Deruyver, and Pierre Parrend. 2018. A systematic survey on multi-step attack detection. *Computers & Security* 76 (2018), 214–249. DOI: <https://doi.org/10.1016/j.cose.2018.03.001>
- [111] Richard Nock and Aditya Krishna Menon. 2020. Supervised learning: No loss no cry. In *Proceedings of the 37th International Conference on Machine Learning (ICML ’20)*. JMLR.org, Article 683, 11 pages. Retrieved from <https://dl.acm.org/doi/pdf/10.5555/3524938.3525621>
- [112] Department of Defense. 2013. Resilient Military Systems and the Advanced Cyber Threat. Retrieved from <https://apps.dtic.mil/sti/citations/ADA569975>
- [113] Department of Defense. 2023. People’s Republic of China State-Sponsored Cyber Actor Living off the Land to Evade Detection. Retrieved from [https://media.defense.gov/2023/May/24/2003229517/-1/-1/0/CSA\\_Living\\_off\\_the\\_Land.PDF](https://media.defense.gov/2023/May/24/2003229517/-1/-1/0/CSA_Living_off_the_Land.PDF)
- [114] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjorn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. 2021. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* 372 (2021), 36. DOI: <https://doi.org/10.1136/bmj.n71>
- [115] Incheon Paik. 2016. Situation awareness based on big data analysis. In *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC ’16)*, Vol. 2. IEEE, 911–916. DOI: <https://doi.org/10.1109/ICMLC.2016.7873008>
- [116] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. 2017. Practical whole-system provenance capture. In *Proceedings of the Symposium on Cloud Computing (SoCC ’17)*. ACM, New York, NY, 405–418. DOI: <https://doi.org/10.1145/3127479.3129249>
- [117] Tejas Patel. 2017. Cyber-Hunting at Scale (CHASES). Retrieved from <https://www.darpa.mil/program/cyber-hunting-at-scale>
- [118] Ramesh Paudel and H. Howie Huang. 2022. Pikachu: Temporal walk based dynamic graph embedding for network anomaly detection. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, Budapest, Hungary, 1–7. DOI: <https://doi.org/10.1109/NOMS54207.2022.9789921>
- [119] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’14)*. ACM, New York, NY, 701–710. DOI: <https://doi.org/10.1145/2623330.2623732>
- [120] yulu Qi, Rong Jiang, Yan Jia, and Aiping Li. 2018. Analysis on the related evaluation technique of composite attack in network confrontation. In *Proceedings of the IEEE Third International Conference on Data Science in Cyberspace (DSC ’18)*. IEEE, 956–962. DOI: <https://doi.org/10.1109/DSC.2018.00152>
- [121] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid. 2021. *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach*. NIST Special Publications, 310 pages. Retrieved from <https://csrc.nist.gov/publications/detail/sp/800-160/vol-2-rev-1/final>
- [122] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. arXiv:2006.10637. Retrieved from <https://doi.org/10.48550/arXiv.2006.10637>
- [123] scikit learn. 2023. Novelty Detection with Local Outlier Factor (LOF). Retrieved November 14, 2023 from [https://scikit-learn.org/stable/auto\\_examples/neighbors/plot\\_lof\\_novelty\\_detection.html](https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_novelty_detection.html)
- [124] Joseph Sexton, Curtis Storlie, and Joshua Neil. 2015. Attack chain detection. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8 (Sep. 2015), 353–363. DOI: <https://doi.org/10.1002/sam.11296>
- [125] Jia Shan-Shan and Xu Ya-Bin. 2017. The APT detection method in SDN. In *Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC ’17)*. IEEE, 1240–1245. DOI: <https://doi.org/10.1109/CompComm.2017.8322741>
- [126] Jia Shan-Shan and Xu Ya-Bin. 2018. The APT detection method based on attack tree for SDN. In *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy (ICCSPP ’18)*. ACM, New York, NY, 116–121. DOI: <https://doi.org/10.1145/3199478.3199481>
- [127] Amit Sharma, Brij B. Gupta, Awadhesh Kumar Singh, and V. K. Saraswat. 2023. Advanced persistent threats (APT): Evolution, anatomy, attribution and countermeasures. *Journal of Ambient Intelligence and Humanized Computing* 14, 7 (May 2023), 9355–9381. DOI: <https://doi.org/10.1007/s12652-023-04603-y>
- [128] Yun Shen, Enrico Mariconti, Pierre Antoine Vervier, and Gianluca Stringhini. 2018. Tiresias: Predicting security events through deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS ’18)*. ACM, New York, NY, 592–605. DOI: <https://doi.org/10.1145/3243734.3243811>
- [129] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021. Masked label prediction: Unified message passing model for semi-supervised classification. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI ’21)*. Zhi-Hua Zhou (Ed.), International Joint Conferences on Artificial Intelligence Organization, Online Only, 1548–1554. DOI: <https://doi.org/10.24963/ijcai.2021/214>

- [130] Xiaokui Shu, Danfeng Yao, and Naren Ramakrishnan. 2015. Unearthing stealthy program attacks buried in extremely long execution paths. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, 401–413. DOI : <https://doi.org/10.1145/2810103.2813654>
- [131] Vaddi Sowmya Sree, Chaitna Sri Koganti, Srinivas K. Kalyana, and P. Anudeep. 2021. Artificial intelligence based predictive threat hunting in the field of cyber security. In *Proceedings of the 2nd Global Conference for Advancement in Technology (GCAT '21)*. IEEE, 1–6. DOI : <https://doi.org/10.1109/GCAT52182.2021.9587507>
- [132] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. 2018. *MITRE ATT-&-CK: Design and Philosophy*. Technical Report. The MITRE Corporation, McLean, VA. Retrieved from <https://www.mitre.org/sites/default/files/2021-11/prs-19-01075-28-mitre-attack-design-and-philosophy.pdf>
- [133] Yunfei Su, Mengjun Li, ChaoJing Tang, and Rongjun Shen. 2015/12. A framework of APT detection based on dynamic analysis. In *Proceedings of the 2015 4th National Conference on Electrical, Electronics and Computer Engineering*. Atlantis Press, 1047–1053. DOI : <https://doi.org/10.2991/nceece-15.2016.187>
- [134] Eytan Tepper. 2022. The First Space-Cyber War and the Need for New Regimes and Policies. Retrieved from [https://www.cigionline.org/static/documents/PB\\_no.173\\_uPqYILM.pdf](https://www.cigionline.org/static/documents/PB_no.173_uPqYILM.pdf)
- [135] S. Thejaswini and C. Indupriya. 2019. Big data security issues and natural language processing. In *Proceedings of the 3rd International Conference on Trends in Electronics and Informatics (ICOEI '19)*. IEEE, 1307–1312. DOI : <https://doi.org/10.1109/ICOEI.2019.8862744>
- [136] threaTrace-detector. 2021. threaTrace. Retrieved November 14, 2023 from <https://github.com/threaTrace-detector/threaTrace>
- [137] Carnegie Mellon University. 2023. The CERT Division—Software Engineering Institute. Retrieved November 14, 2023 from <https://www.sei.cmu.edu/about/divisions/cert/>
- [138] Jesper E. van Engelen and Holger H. Hoos. 2019. A survey on semi-supervised learning. *Machine Learning* 109, 2 (Nov. 2019), 373–440. DOI : <https://doi.org/10.1007/s10994-019-05855-6>
- [139] Petar Veličković. 2023. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology* 79 (2023), 102538. DOI : <https://doi.org/10.1016/j.sbi.2023.102538>
- [140] Anh V. Vu, Daniel R. Thomas, Ben Collier, Alice Hutchings, Richard Clayton, and Ross Anderson. 2022. Getting bored of cyberwar: Exploring the role of civilian participation in the Russia-Ukraine cyber conflict. arXiv:2208.10629. Retrieved from <https://doi.org/10.48550/ARXIV.2208.10629>
- [141] Jan Vykopal, Pavel Celeda, Pavel Seda, Valdemar Svabensky, and Daniel Tovarnak. 2021. Scalable learning environments for teaching cybersecurity hands-on. In *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE, Lincoln, NE, 1–9. DOI : <https://doi.org/10.1109/fie49875.2021.9637180>
- [142] Jan Vykopal, Martin Vizvary, Radek Oslajsek, Pavel Celeda, and Daniel Tovarnak. 2017. Lessons learned from complex hands-on defence exercises in a cyber range. In *Proceedings of the IEEE Frontiers in Education Conference (FIE '17)*. IEEE, 1–8. DOI : <https://doi.org/10.1109/fie.2017.8190713>
- [143] Qi Wang, Wajih Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl. A. Gunter, and Haifeng Chen. 2020. You are what you do: Hunting stealthy malware via data provenance analysis. In *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2020*. Internet Society. DOI : <https://doi.org/10.14722/ndss.2020.24167>
- [144] Su Wang, Zhiliang Wang, Tao Zhou, Hongbin Sun, Xia Yin, Dongqi Han, Han Zhang, Xingang Shi, and Jiahai Yang. 2022. THREATTRACE: Detecting and tracing host-based threats in node level through provenance graph learning. *IEEE Transactions on Information Forensics and Security* 17 (2022), 3972–3987. DOI : <https://doi.org/10.1109/TIFS.2022.3208815>
- [145] Yuan Wang, Yongjun Wang, Jing Liu, and Zhijian Huang. 2014. A network gene-based framework for detecting advanced persistent threats. In *Proceedings of the 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, 97–102. DOI : <https://doi.org/10.1109/3PGCIC.2014.41>
- [146] Renzheng Wei, Lijun Cai, Lixin Zhao, Aimin Yu, and Dan Meng. 2021. DeepHunter: A graph neural network based approach for robust cyber threat hunting. In *Security and Privacy in Communication Networks*. Joaquin Garcia-Alfaro, Shujun Li, Radha Poovendran, Hervé Debar, and Moti Yung (Eds.), Springer International Publishing, 3–24. DOI : [https://doi.org/10.1007/978-3-030-90019-9\\_1](https://doi.org/10.1007/978-3-030-90019-9_1)
- [147] B. Weisfeiler and A. Lehman. 1968. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Journal of Applied Mathematics and Physics*. Nauchno-Technicheskaya Informatsia. Retrieved from [https://www.iti.zcu.cz/wl2018/pdf/wl\\_paper\\_translation.pdf](https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf)
- [148] Senhao Wen, Nengqiang He, and Hanbing Yan. 2017. Detecting and predicting APT based on the study of cyber kill chain with hierarchical knowledge reasoning. In *Proceedings of the 2017 VI International Conference on Network, Communication and Computing (ICNCC '17)*, Vol. 17, 115–119. DOI : <https://doi.org/10.1145/3171592.3171641>
- [149] Senhao Wen, Yu Rao, and Hanbing Yan. 2018. Information protecting against APT based on the study of cyber kill chain with weighted Bayesian classification with correction factor. In *Proceedings of the 7th International Conference on Informatics, Environment, Energy and Applications (IEEA '18)*. ACM, New York, NY, 231–235. DOI : <https://doi.org/10.1145/3208854.3208893>

- [150] Florian Wilkens, Felix Ortmann, Steffen Haas, Matthias Vallentin, and Mathias Fischer. 2021. Multi-stage attack detection via kill chain state machines. In *Proceedings of the 3rd Workshop on Cyber-Security Arms Race (CYSARM '21)*. ACM, New York, NY, 13–24. DOI : <https://doi.org/10.1145/3474374.3486918>
- [151] Jeannette M. Wing. 2006. Attack graph generation and analysis. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security (ASLACCS '06)*. ACM, New York, NY, 14. DOI : <https://doi.org/10.1145/1128817.1128822>
- [152] Jun Wu, Kaoru Ota, Mianxiong Dong, and Chunxiao Li. 2016. A hierarchical security framework for defending against sophisticated attacks on wireless sensor networks in smart cities. *IEEE Access* 4 (2016), 416–424. DOI : <https://doi.org/10.1109/ACCESS.2016.2517321>
- [153] Bin Xia, Junjie Yin, Jian Xu, and Yun Li. 2019. LogGAN: A sequence-based generative adversarial network for anomaly detection based on system logs. In *Science of Cyber Security*. Feng Liu, Jia Xu, Shouhuai Xu, and Moti Yung (Eds.), Springer International Publishing, 61–76. DOI : [https://doi.org/10.1007/978-3-030-34637-9\\_5](https://doi.org/10.1007/978-3-030-34637-9_5)
- [154] Liang Xiao, Dongjin Xu, Narayan B. Mandayam, and H. Vincent Poor. 2018. Attacker-centric view of a detection game against advanced persistent threats. *IEEE Transactions on Mobile Computing* 17, 11 (2018), 2512–2523. DOI : <https://doi.org/10.1109/TMC.2018.2814052>
- [155] Yulai Xie, Dan Feng, Yuchong Hu, Yan Li, Staunton Sample, and Darrell Long. 2020. Pagoda: A hybrid approach to enable efficient real-time provenance based intrusion detection in big data environments. *IEEE Transactions on Dependable and Secure Computing* 17, 6 (2020), 1283–1296. DOI : <https://doi.org/10.1109/TDSC.2018.2867595>
- [156] Yulai Xie, Dan Feng, Zhipeng Tan, and Junzhe Zhou. 2016. Unifying intrusion detection and forensic analysis via provenance awareness. *Future Generation Computer Systems* 61 (2016), 26–36. DOI : <https://doi.org/10.1016/j.future.2016.02.005>
- [157] Chunlin Xiong, Tiantian Zhu, Weihao Dong, Linqi Ruan, Runqing Yang, Yueqiang Cheng, Yan Chen, Shuai Cheng, and Xutong Chen. 2022. Conan: A practical real-time APT detection system with high accuracy and efficiency. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (2022), 551–565. DOI : <https://doi.org/10.1109/TDSC.2020.2971484>
- [158] Cho Do Xuan and Mai Hoang Dao. 2021. A novel approach for APT attack detection based on combined deep learning model. *Neural Computing and Applications* 33, 20 (Apr. 2021), 13251–13264. DOI : <https://doi.org/10.1007/s00521-021-05952-5>
- [159] Feng Xuewei, Wang Dongxia, Huang Minhuan, and Sun Xiaoxia. 2014. An approach of discovering causal knowledge for alert correlating based on data mining. In *Proceedings of the IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*. IEEE, 57–62. DOI : <https://doi.org/10.1109/DASC.2014.19>
- [160] Muhammad Mudassar Yamin, Basel Katt, and Vasileios Gkioulous. 2020. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. *Computers & Security* 88 (2020), 101636. DOI : <https://doi.org/10.1016/j.cose.2019.101636>
- [161] Dingqi Yang, Bin Li, Laura Rettig, and Philippe Cudré-Mauroux. 2017. HistoSketch: Fast similarity-preserving sketching of streaming histograms with concept drift. In *Proceedings of the IEEE International Conference on Data Mining (ICDM '17)*. IEEE, 545–554. DOI : <https://doi.org/10.1109/ICDM.2017.64>
- [162] Jian Yang, Qi Zhang, Xiaofeng Jiang, Shuangwu Chen, and Feng Yang. 2022. Poirot: Causal correlation aided semantic analysis for advanced persistent threat detection. *IEEE Transactions on Dependable and Secure Computing* 19, 5 (2022), 3546–3563. DOI : <https://doi.org/10.1109/TDSC.2021.3101649>
- [163] Han Yu, Aiping Li, and Rong Jiang. 2019. Needle in a haystack: Attack detection from large-scale system audit. In *Proceedings of the IEEE 19th International Conference on Communication Technology (ICCT '19)*. IEEE, 1418–1426. DOI : <https://doi.org/10.1109/ICCT46805.2019.8947201>
- [164] Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 2018. NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, 2672–2681. DOI : <https://doi.org/10.1145/3219819.3220024>
- [165] Jun Zeng, Zheng Leong Chua, Yinfang Chen, Kaihang Ji, Zhenkai Liang, and Jian Mao. 2021. WATSON: Abstracting behaviors from audit logs via aggregation of contextual semantics. In *Proceedings 2021 Network and Distributed System Security Symposium*. Internet Society, Online Only, 18. DOI : <https://doi.org/10.14722/ndss.2021.24549>
- [166] Jun Zengyi, Xiang Wang, Jiahao Liu, Yinfang Chen, Zhenkai Liang, Tat-Seng Chua, and Zheng Leong Chua. 2022. SHADEWATCHER: Recommendation-guided cyber threat analysis using system audit records. In *Proceedings of the IEEE Symposium on Security and Privacy (SP '22)*. IEEE, 489–506. DOI : <https://doi.org/10.1109/SP46214.2022.9833669>
- [167] Xu Zhang, Yong Xu, Qingwei Lin, Bo Qiao, Hongyu Zhang, Yingnong Dang, Chunyu Xie, Xinsheng Yang, Qian Cheng, Ze Li, Junjie Chen, Xiaoting He, Randolph Yao, Jian-Guang Lou, Murali Chintalapati, Furoo Shen, and Dongmei Zhang. 2019. Robust log-based anomaly detection on unstable log data. In *Proceedings of the 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '19)*. ACM, New York, NY, 807–817. DOI : <https://doi.org/10.1145/3338906.3338931>
- [168] Chaofan Zheng, Wenhui Hu, Tianci Li, Xueyang Liu, Jinchan Zhang, and Litian Wang. 2022. An insider threat detection method based on heterogeneous graph embedding. In *Proceedings of the IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS '22)*. IEEE, 11–16. DOI : <https://doi.org/10.1109/BigDataSecurityHPSCIDS54978.2022.00013>
- [169] Aaron Zimba, Hongsong Chen, Zhaoshun Wang, and Mumbi Chishimba. 2020. Modeling and detection of the multi-stages of advanced persistent threats attacks based on semi-supervised learning and complex networks characteristics. *Future Generation Computer Systems* 106, C (May 2020), 501–517. DOI : <https://doi.org/10.1016/j.future.2020.01.032>

- [170] Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. 2022. Provenance-based intrusion detection systems: A survey. *ACM Computing Surveys* 55, 7 (May 2022), 36 pages. DOI: <https://doi.org/10.1145/3539605>
- [171] Qingtian Zou, Anoop Singhal, Xiaoyan Sun, and Peng Liu. 2020. Automatic recognition of advanced persistent threat tactics for enterprise security. In *Proceedings of the 6th International Workshop on Security and Privacy Analytics (IWSPA '20)*. ACM, New York, NY, 43–52. DOI: <https://doi.org/10.1145/3375708.3380314>

Received 31 January 2024; revised 30 July 2024; accepted 29 August 2024