Traffic & Transportation

Proactive Detection and Mitigation Strategies for Advanced Persistent Threats

Raghav MITTAL[1], Ivan CVITIĆ[2], Dragan PERAKOVIĆ[3], Soosaimarian Peter RAJA[4]

Original Scientific Paper
Submitted: 7 Feb 2025
Accepted: 31 Mar 2025

[1] raghav.mittal2021a@vitstudent.ac.in, Vellore Institute of Technology, School of Computer Science and Engineering
[2] icvitic@fpz.unizg.hr, University of Zagreb, Faculty of Transport and Traffic Sciences
[3] dperakovic@fpz.unizg.hr, University of Zagreb, Faculty of Transport and Traffic Sciences
[4] avemariaraja@gmail.com, Vellore Institute of Technology, School of Computer Science and Engineering

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
This research explores the growing threat of advanced persistent threats (APTs), which pose significant risks to national security, organisational operations and critical infrastructure. APTs have become increasingly sophisticated, targeting various sectors and demanding more effective defences to protect sensitive data and key systems. The focus of this paper is on addressing the rising frequency and complexity of APT attacks, aiming to provide a detailed analysis of their evolving tactics and the need for proactive security measures. Specifically, the paper examines current gaps in APT detection, from the initial stages of infiltration through to the complete removal of the threat. To address these challenges, the study introduces several detection strategies, including advanced correlation techniques, behavioural analysis of network traffic and user activity, and the application of machine learning and AI to improve threat identification. The paper analyses real-world APT incidents and discusses how monitoring and deception tactics can enhance security measures. It highlights the ongoing challenges presented by APTs, particularly their adaptive and dynamic attack methods, and emphasises the need for continuous improvement in defensive strategies. In conclusion, the paper outlines key areas for future research and stresses the importance of a proactive, evolving approach to counter the persistent and evolving nature of APTs.

KEYWORDS
advanced persistent threats; Stuxnet; Nash equilibrium; game theory; online adaptive metric learning; hidden Markov model; Carbanak; Hydraq.

1. INTRODUCTION

Cyberattacks have emerged as the most serious risk to cybersecurity in recent years. Because of their extremely destructive nature, advanced persistent threats (APTs) have received a lot of attention among these threats. APTs deliberately target governments and businesses around the world through cyber espionage and criminal activity. Rather than creating immediate damage, these assaults concentrate on monitoring system activity and continuously obtaining sensitive data. As a result, the loss of such important information can result in reduced service quality, severe security incidents, and serious ramifications for secret government agencies and security-sensitive systems. Due to these reasons, this research paper has been aimed at understanding advanced persistent threats in detail while analysing and comparing the various detection methods for the same.

APT attackers use sophisticated tools and strategies to circumvent standard security measures such as signature-based antivirus software and intrusion detection and prevention systems (IDS/IPS). They spend time researching the target organisation's infrastructure, applications and vulnerabilities in order to generate

customised malware that can avoid detection. To get initial network access, spear-phishing and other social engineering techniques are frequently utilised. The persistent nature of APT attacks, as well as their ability to exploit undisclosed weaknesses, underline the critical need for robust defence methods [1].

APT assaults are distinguished by their multi-stage nature, which includes extensive surveillance and precise targeting. The goal of an attacker is to acquire unauthorised access to a victim's system and escalate privileges to the root level. Additionally, they then use information flow pathways to extract sensitive data [2]. APT campaigns such as Operation Aurora, Duqu, Flame and Fox Kitten have shown the worldwide reach and deadly consequences of these attacks on critical infrastructure and information theft. APTs constantly modify their plans and approaches to avoid detection, emphasising the importance of defenders understanding and countering their developing tactics [3].

Through multiple methods of reconnaissance, the APT scopes the target. Then, through the initial attack vector, the APT gains access to the system. Following this through techniques like privilege escalation and lateral movement, the APT increases its access and reach. Finally, the APT gains access to the sensitive information server in order to get crucial data.

Effective identification of APT assaults is critical for limiting their damage and safeguarding sensitive data. Existing solutions take one of two ways. The first approach identifies APT information flows using machine learning classification approaches such as one-class classifiers or decision trees [4]. The second technique employs game theory models to represent the interaction between APTs and defenders, with the goal of determining the Nash equilibrium [5]. These approaches, however, confront hurdles due to feature scarcity, difficulty in capturing novel assault patterns, and limited collaboration between game theory and machine learning methods.

In this research, three primary methodologies have been selected for in-depth analysis: game theory, machine learning and pattern matching. The decision to focus on these methodologies is driven by their proven efficacy and widespread utilisation in the field of advanced threat detection.

Within game theory, the Nash equilibrium model is chosen due to its robust application in strategising defence mechanisms against adversarial threats. For machine learning, the online adaptive metric learning (OAML) approach is selected, given its capacity to adaptively and dynamically learn from data, which is crucial for detecting evolving threats in real-time. Lastly, the hidden Markov model (HMM) is identified from the pattern matching techniques, owing to its effectiveness in modelling sequences of observed events and uncovering underlying patterns indicative of advanced persistent threats.

This selection aims to leverage the strengths of each methodology, providing a comprehensive review of the most effective and widely adopted models in advanced persistent threat detection. By examining these techniques, the research contributes to the literature by offering a thorough assessment of their applicability and effectiveness in detecting APTs.

This study makes several key contributions to the field of advanced persistent threat (APT) detection. Unlike previous research that primarily focuses on detecting APTs at specific stages of an attack, this paper presents a holistic approach that spans the entire attack lifecycle, from initial reconnaissance to post-attack remediation. It introduces a comparative evaluation of three advanced detection methodologies – Nash equilibrium (game theory), online adaptive metric learning (machine learning) and hidden Markov models (pattern matching) – assessing their effectiveness in identifying and mitigating APT threats. The study further highlights the advantages and limitations of each model, providing a detailed analysis of their applicability in real-world scenarios. Additionally, it addresses the integration challenges between game-theoretic strategies and machine learning-based detection, offering insights into how hybrid approaches can improve APT identification. By emphasising the need for adaptive, proactive defence mechanisms, this research contributes to the development of more resilient cybersecurity frameworks capable of countering the evolving nature of APT attacks.

This research paper studies and evaluates the various data, surveys and existing research papers to collate effective solutions and techniques for APT detection and defence. This paper is organised as follows: Section 2 provides an overview of past APT attacks and malware. Additionally, it analyses past research into APT detection mechanisms and tools, providing the basis for this paper. Section 3 takes a deep dive into the methodologies used to analyse the multiple detection techniques for APT. Section 4 takes a closer look at the performance evaluation and how the three different models compare to each other, alongside research result analysis. Section 5 discusses the identified challenges pertaining to the research of detection techniques for advanced persistent threats. Section 6 identifies further research opportunities in the field of APT detection.

Finally, Section 7 concludes the paper, summarises the spectrum of studies that have been discussed and outlines future directions of research.

2. PREVIOUS RESEARCH

In this section of the paper, an overview of the current and past research about advanced persistent threat attacks, as well as detection techniques, will be provided. Further, the section will be divided into two broad classifications, namely, analysis of past APT attacks and detection of APT attacks.

2.1 Analysis of past APT attacks

For the purposes of maintaining uniformity in comparison, a certain set of metrics is being used to study the case studies of APT-based malware. These metrics include:

— Target entity, which includes government institutes or organisations, private corporations, data storage centres, etc.
— Initial infection vector, which pertains to the medium through which the virus/worm/malware was introduced to the host system.
— Malware analysis: It is the study of the techniques the malware utilises to spread and affect the host network or database.
— Infrastructure: It is the infrastructure, architecture and organisation of the malware.
— Lateral movement and proliferation: These include the tactics the malware employs in order to achieve privilege escalation and access to restricted information.
— Exfiltration: These involve the methods used by malware in order to transfer stolen information from the compromised host back to the attacker.
— Evasion and anti-forensic techniques: These include the strategies used by the malware in order to conceal itself and evade anti-malware procedures such as anti-virus software, periodic system scans, etc.
— Lessons learned: This last metric pertains to the understanding of how to detect and protect the system from a particular malware by understanding the shortcomings and gaps in the protection systems.

The Hydraq APT, also known as Aurora or Operation Aurora, initiated a complex cyber-attack in 2009 targeting major technology firms using advanced spear-phishing and social engineering to infiltrate networks. The primary tool utilised by Hydraq was a sophisticated remote access trojan (RAT) equipped with evasion techniques such as code obfuscation, anti-virtualisation and anti-debugging capabilities to dodge standard security measures. To maintain control and persistence within the compromised networks, the APT used a multifaceted command and control (C2) infrastructure spread across different jurisdictions, employing encrypted communication protocols and custom encryption to complicate tracking and analysis.

Once inside the targeted systems, Hydraq actors executed extensive lateral movement and privilege escalation using compromised credentials, exploited vulnerabilities and targeted weak security configurations. This allowed them to access critical systems and extract valuable intellectual property and sensitive data covertly, utilising tactics like encrypted channels, steganography and data compression to avoid detection. The operation underscored the importance of proactive threat hunting, continuous security monitoring, and the need for multi-layered defence strategies to effectively counteract such advanced persistent threats. By dissecting the TTPs employed in the Hydraq attacks, organisations can enhance their defence mechanisms to better detect, prevent and respond to future APT incidents.

The Carbanak APT group, known for its financial attacks on banking institutions, employs various methods for initial compromise, including spear-phishing, watering hole attacks and exploit kits, using sophisticated social engineering tactics to deploy malware like the Anunak/Carbanak RAT. This malware facilitates persistent system access, command and control (C2) operations through encrypted channels like VPNs or Tor, and data exfiltration, making it difficult for defenders to intercept or analyse malicious communications. Carbanak's strategy focuses on maintaining stealth and operational security within infiltrated networks, enabling them to explore and identify high-value targets crucial for later stages of their financial theft operations.

Once established within a target network, Carbanak emphasises lateral movement and persistence, using techniques such as password cracking, exploitation of vulnerable configurations and leveraging legitimate tools like PowerShell. Their ultimate objective is financial gain through sophisticated manipulation of financial systems, including abuse of the SWIFT network, unauthorised transactions and ATM network compromises. Additionally, Carbanak employs various anti-forensic methods such as log deletion, encrypted

communications and timestamp manipulation to hinder forensic investigations and evade attribution. These evolving tactics underscore the need for robust cybersecurity measures, continuous vigilance and proactive threat hunting to mitigate the advanced and adaptive threats posed by Carbanak and similar financially motivated APT groups [6].

Stuxnet, an exemplary instance of an advanced persistent threat (APT), utilised diverse infection vectors, including exploiting zero-day vulnerabilities and employing social engineering tactics, to gain initial access to targeted systems. Its sophisticated modular design, comprising rootkit functionality and PLC payload, enabled precise manipulation of Siemens industrial control systems, particularly impacting Iran's nuclear program. Stuxnet's complex command and control (C2) infrastructure, employing peer-to-peer networks and compromised legitimate websites, facilitated stealthy communication and updates from its handlers, while its exploitation of zero-day vulnerabilities bypassed security safeguards, highlighting its advanced capabilities.

Once inside a network, Stuxnet employed advanced lateral movement techniques, leveraging stolen digital certificates and exploiting vulnerable systems to proliferate and manipulate targeted industrial control systems, notably disrupting centrifuges in Iran's nuclear enrichment plants. To evade detection and analysis, Stuxnet employed sophisticated evasion and anti-forensic techniques, including rootkit features and encrypted communication channels, showcasing the formidable challenges posed by APTs in critical infrastructure settings. Lessons learned from the Stuxnet case underscore the necessity for comprehensive defence-in-depth strategies, continual monitoring, secure supply chain practices and international collaboration to mitigate the evolving threats of APTs targeting critical systems [7].

2.2 Detection of APT attacks

There has been extensive research into individual models and tools for APT detection. These techniques employ varying strategies from artificial intelligence, machine learning, pattern matching, graph analysis, anomaly detection and many more.

Detecting advanced persistent threats (APTs) remains a significant challenge in current intrusion detection systems (IDSs). Researchers have devoted considerable effort to tackling this issue. One innovative solution proposed in [8], named "SPuNgae", is a host-based APT detector that monitors network activity to identify malicious URLs. Sigholm et al. in [9] proposed using a data leakage prevention (DLP) algorithm to detect data exfiltration, with cyber counterintelligence (CCI) sensors accurately pinpointing the location of leaked data. Another method, TerminAPT, discussed by Brogi et al. in [10], tracks data flow within APT campaigns.

APT campaigns often begin with spear phishing attacks to gain entry. In [11], Chandra et al. explored mathematical computational techniques for analysing spam emails to identify spam behaviour. While effective, this method's limitation lies in thoroughly defining tokens in the algorithm.

Several APT detection and prevention models, including machine learning-based methods, have been developed, but many have drawbacks in terms of accuracy, precision and recall, as discussed in subsequent sections.

Renowned cybersecurity researchers have devised various approaches to handle APT attacks. For instance, Bari in [12], the honey-pot technique is suggested for deployment within the network infrastructure. However, it lacks real-time APT detection and post-infiltration detection capabilities.

Big data analytics, as discussed by Cardenas et al. in [13], has emerged as a promising technique for APT detection, leveraging pattern matching and network topology analysis. Despite its advantages, it may suffer from false positives and lack real-time protection.

Lastly, a context-based framework introduced by Giura et al. in [14] aims to enhance user experience in information systems, particularly in medical or context-based environments.

In the further sections, this paper explores a few APT detection mechanisms and compares the various tools to provide a comparative analysis and conclude which technique provides the most efficient APT detection.

3. METHODOLOGY

3.1 Approach 1 (Nash equilibrium – game theory)

The concept of Nash equilibrium in the context of the APT detection technique refers to a stable state where both the attacker and the defender have chosen their optimal strategies, and neither has an incentive to unilaterally deviate from their chosen strategy. In other words, it represents a state where neither the attacker nor the defender can improve their outcome by changing their strategy while the other remains unchanged.

In APT scenarios, the attacker aims to compromise system resources, while the defender's goal is to protect those resources and minimise the damage caused by the attacker. The Nash equilibrium in APT detection techniques is achieved by finding the optimal strategies for both the attacker and the defender [15].

The attacker's strategy is determined by factors such as the attack rate, cost of launching attacks and potential gains from compromising resources. The attacker aims to minimise their cost while maximising their gain from the compromised resources. Similarly, the defender's strategy involves factors such as the recapture rate, the cost of recapturing compromised resources and the overall damage caused by the compromise. The defender aims to minimise the cost of recapture while minimising the damage caused by the compromise [16].

The Nash equilibrium can be reached by solving optimisation issues by mathematically modelling the cost functions for both the attacker and the defender. The equilibrium point represents the ideal strategies for both the attacker and the defender, ensuring that neither party may improve their outcome unilaterally by changing their plan while the other player retains their optimal strategy.

An APT detection technique can find the ideal strategy for both the attacker and the defender by determining the Nash equilibrium, providing insights into the attacker's behaviour, and allowing the defence to create effective countermeasures to safeguard system resources [17].

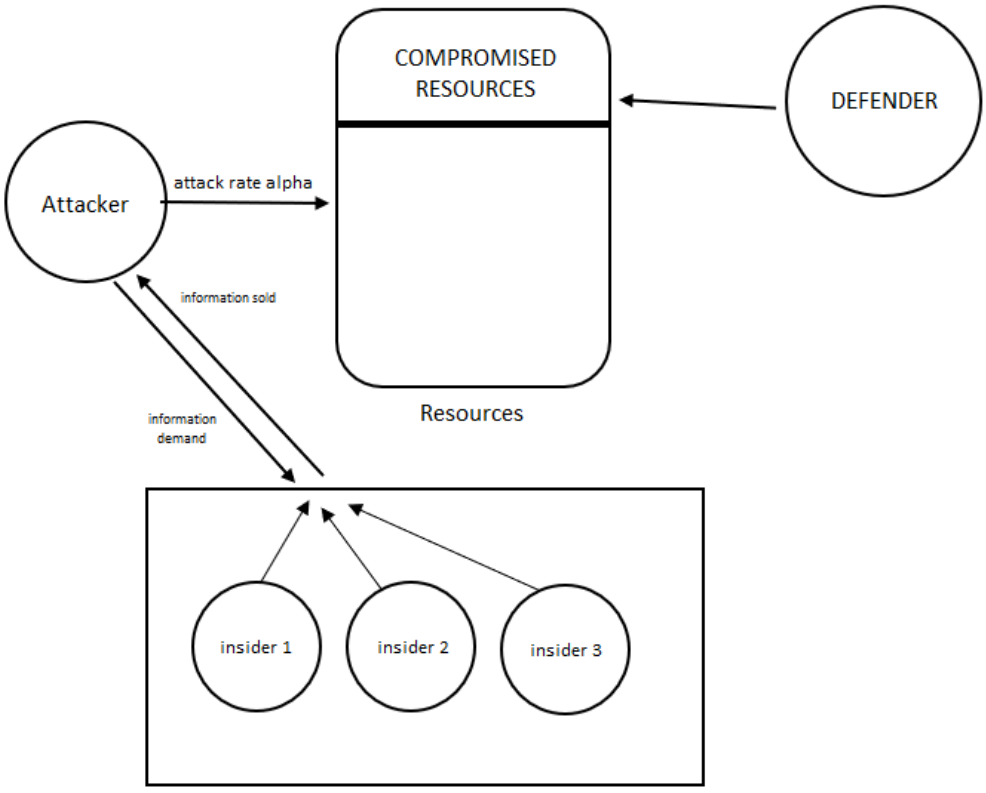

Figure 1 – Nash equilibrium based model for APT detection [17]

In *Figure 1*, we look at a system that is under attack from both an APT attacker and insiders. The system is made up of four major parts: the system resource, an APT attacker, a defender and n insiders. The system resource, which includes the firewall, network, software and operating system, is the major target of these threats. The above figure depicts the interaction between the defence, APT attacker and insiders, demonstrating the dynamic interplay between these entities.

In *Algorithm 1*, the findNashEquilibrium function takes the unit costs for the attacker's secure resource (rA), risk of being caught by the attacker (qA), defender's compromised resources (rD) and defender's recapture rate (qD) as input. It determines the Nash equilibrium by evaluating the conditions stated in Theorem 1 and returns the optimal values for the attacker's strategy (alpha_star) and defender's strategy (beta_star).

One can assign the specific values for rA, qA, rD and qD in the example usage section to obtain the Nash equilibrium for your specific APT detection scenario based on the static model.

Algorithm 1 – APT Detection using Nash equilibrium model

```
function findNashEquilibrium(rA, qA, rD, qD):
    if rA/qA < rD/qD < 1:
        alpha_star = rA/qA
        beta_star = 1
    else if rD/qD < rA/qA < 1:
        alpha_star = 1
        beta_star = rD/qD
    else if rA/qA = rD/qD < 1:
        alpha_star = beta_star = rA/qA
    else:
        alpha_star = beta_star = 1

    return alpha_star, beta_star

# Example usage:
rA = <value>  # Unit cost for attacker's secure resource
qA = <value>  # Unit cost for risk of being caught by the attacker
rD = <value>  # Unit cost for defender's compromised resources
qD = <value>  # Unit cost for defender's recapture rate

alpha_star, beta_star = findNashEquilibrium(rA, qA, rD, qD)
```

This section proposes a general model of the combination of APT and insider threat, which treats the interaction between the defence, APT attacker and many insiders as a two-layer differential game. The threat's purpose is to compromise the system resources. The APT attacker intends to acquire harmful gains by launching assaults and compromising resources, sometimes with the assistance of inside information obtained from insiders. Insiders maximise their financial gains by selling inside information to the APT attacker. The aim of the defender is to recover compromised resources and limit the harm caused by APT and insider attacks [18].

A system model is presented, where the total system resource is normalised to the value of 1. The fraction of compromised resources at time t is denoted by x(t), ranging from 0 (fully protected system) to 1 (completely compromised system). The actions of the APT attacker and the defender modify the system state as per the dynamics given by:

$$\dot{x}(t) = \alpha \cdot \big(1 - x(t)\big) - \beta \cdot x(t) \tag{1}$$

where α represents the attack rate of the APT attacker and β is the recapture rate of the defender. The evolving rate of the system state is determined by the percentages of resources seized by the attacker and recaptured by the defender.

The cost model considers the expenditures experienced by the attacker and the defender over a long period of time. The cost of the attacker is made up of the danger of being identified by the defender and the utility gain from compromised resources. The attacker's instantaneous cost is as follows:

$$cA(x(t), \alpha, \beta, t) = rA\big(1 - x(t)\big)^2 + qA\alpha^2\big(1 - x(t)\big)^2 \tag{2}$$

The unit costs, represented as rA and qA, are positive constant numbers that represent the costs associated with the secure resource and, respectively, the danger of being caught. The secure resource's instantaneous cost function is determined by the system's state, as illustrated by the first part of *Equation (2)*, which is rA multiplied by the squared difference between 1 and $x(t)$. The term $qA2\big(1 - x(t)\big)^2$, on the other hand, reflects the estimated cost of being discovered by the defender while launching attacks at time t. These costs are captured using a quadratic cost model. Similarly, the cost of the defender comprises both the operational cost of scanning and recapturing compromised resources and the damage inflicted by these resources. The defender's instantaneous cost is calculated as follows:

$$cD(x(t), \alpha, \beta, t) = rDx(t)^2 + qD\beta^2 x(t)^2 \tag{3}$$

The unit costs, denoted by rD and qD, are constants. The cost of compromised resources, given as cD in *Equation (3)*, is determined by system state and is proportional to the squared value of $x(t)$ with a coefficient of rD. Furthermore, the word $qD\beta^2 x(t)^2$ represents the cost suffered when the defender begins efforts to recapture the compromised resources.

It is typical for attackers to acquire a foothold within a system for future attacks in the context of an advanced persistent threat (APT) scenario. Given that APT attackers are well-funded, and insiders are motivated by monetary gain, an attacker's strategic strategy is to obtain confidential information, such as passwords, from insiders to aid in their attacks. At time t, the total information required by the attacker denoted as $Q(t)$, is defined by the attacker's attack rate (t), where $Q(t)$ is a function of (t), represented by $f(\alpha(t))$. The function f() is a non-decreasing general model with regard to (t). At time t, each insider i sells knowledge, indicated by $ui(t)(with\ ui(t) \geq 0)$. The nominal price of information at time t, as per the linear inverse demand function, is calculated as:

$$pb(t) = A(Q(t)) - \Sigma ui(t) \tag{4}$$

from $i = 1$ to n, where $A(.)$ is a non-decreasing function of $Q(t)$ and n is the number of system insiders. pb(t) indicates the nominal price, however, it does not reflect the current market price at which insider knowledge is traded. Due to real-time adjustments in information demand $(Q(t))$ and the availability of information from insiders $(\Sigma ui(t))$, market prices tend to be sticky and do not quickly converge to the nominal price. The market price, represented as $p(t)$, evolves according to the equation's dynamics;

$$p^{\cdot}(t) = s(pb(t) - p(t)) \tag{5}$$

where s is a constant value determining the convergence speed to the nominal price.

Insiders stand the danger of being identified while selling information to the attacker since the defence regularly monitors the system. Each insider's instantaneous risk (cost) function, as given by the equation:

$$C(ui(t)) = cui(t) + ½ * ui(t)2\,, \forall i \in [1, n] \tag{6}$$

Equation (6) includes the unit cost (c) as well as a quadratic element relating to the amount of information sold $(ui(t))$. The defender's scan approach influences the unit cost, meaning that a more active detection strategy raises the risk involved with selling information to insiders.

The revenue from trading inside information is represented as $p(t)ui(t)$, which gives the instantaneous monetary benefit for each insider. As a result, the net profit of insider i at time t, indicated by $i(t)$, is calculated as follows:

$$\pi i(t) = p(t)ui(t) - (cui(t) + ½ * ui(t)2\,) \tag{7}$$

In conclusion, this section offers a general model that describes the interaction between APT attackers, insiders and the defence in a system under joint threat. It describes the system model, system state dynamics, cost models for attackers and defenders and profit models for insiders.

We now look at the solution to the static defence/attack game between the APT attacker and defender. Consider the following scenario: the rates of actions are constant and pre-configured prior to system deployment.

The attacker's goal is to take as many resources as possible while minimising the long-term cost of the attack. The attacker's cost function, as given by *Equation (8)*, can be expressed as follows:

$$JA(\alpha, \beta) = lim(T \to \infty)\left(\frac{1}{T}\right) \int [0, T] cA(x(t), \alpha, \beta, t) dt \tag{8}$$

similarly, the defender aims to protect the resources from the attacker and minimise the cost of recapturing compromised resources. The cost function of the defender, as stated in *Equation (9)*, can be expressed as:

$$JD(\alpha, \beta) = lim(T \to \infty)\left(\frac{1}{T}\right) \int [0, T] cD(x(t), \alpha, \beta, t) dt. \tag{9}$$

Taking into account these cost functions, we can define the Nash equilibrium as follows.

Definition 1: The system dynamics (1) and the cost functions (8) and (9) define the attacker-defender game in which a set of strategies $\{\alpha *, \beta *\}$ establishes a Nash equilibrium only under the condition:

$$JA(\alpha *, \beta *) \leq JA(\alpha, \beta *), \tag{10}$$

$$JD(\alpha *, \beta *) \leq JD(\alpha *, \beta) \tag{11}$$

We assume in the dynamic system that the defender has complete control of the system at the start $(x(0) = 0)$. *Equation (1)* has a generic solution since it is a first-order nonhomogeneous differential equation with constant coefficients. Solving the equation yields the system state at time t, which is given by:

$$x(t) = \frac{\alpha}{(\alpha + \beta)(1 - e^{-(\alpha+\beta)t})} \tag{12}$$

Substituting x(t) into *Equation (8)*, the cost function of the attacker can be derived as:

$$CA(\alpha) = \left(rA + qA * \alpha^2\right)\left(\frac{\beta}{\alpha + \beta}\right)^2 \tag{13}$$

The attacker seeks to minimise its cost function by optimal action, resulting in the following optimisation problem:

$$\begin{matrix} \min \alpha \, CA(\alpha) \\ s.t. \alpha \in [0,1] \end{matrix} \tag{14}$$

Optimal control for the attacker can be determined by means of the following lemma: Lemma 1: The optimal strategy for the attacker in response to the defender is given by

$$\alpha^* = \begin{Bmatrix} \frac{r_A}{q_A.\beta} & r_A/q_A \leq \beta, \\ 1 & r_A/q_A > \beta. \end{Bmatrix} \tag{15}$$

Proof: Taking the derivative of *CA(α) w.r.t. α*,

$$d\frac{CA}{d\alpha} = \frac{(2\beta^2)(qA \cdot \alpha \cdot \beta - rA)}{(\alpha + \beta)^3} \tag{16}$$

Evaluating it at α = 0, we find that *dCA/dα* is negative. Hence, there are two possibilities for *dCA/dα*: it remains negative, or it intersects the horizontal axis once at *α = rA/(qAβ)*. In the first case, *CA(α)* is non-increasing, and its minimum occurs at *α = 1*. In the second case, if *rA/qA ≤ β, CA(α)* is minimised at *α = rA/(qAβ);* otherwise, it is minimised at *α = 1*.

Similarly, we find out the best strategy for the defender through the following lemma:

Lemma 2: The best strategy for the defender in response to the attacker is given by.

$$\beta * = \begin{Bmatrix} \frac{rD}{qD * \alpha} & \frac{rD}{qD} \leq \alpha \\ 1 & \frac{rD}{qD} > \alpha. \end{Bmatrix} \tag{17}$$

On the basis of Definition 1 and Lemmas 1 and 2, the following theorem can be used to prove the presence of a Nash equilibrium in the static case:

Theorem 1: Nash equilibrium can be divided into 4 categories for the static case:
(1) $\alpha * = rA/qA \; and \; \beta * = 1, when \; rA/qA < rD/qD < 1$;
(2) $\alpha * = 1 \; and \; \beta * = rD/qD, when \; rD/qD < rA/qA < 1$;
(3) $\alpha * \; and \; \beta * \; are \; on \; the \; curve \; \alpha * \cdot \beta * = r/q, when \; rA/qA = rD/qD = r/q < 1$; *and*
(4) $\alpha * = 1 \; and \; \beta * = 1, when \; rD/qD > 1 \; and \; rA/qA > 1$.

Proof: The definition of Nash equilibrium is the juncture of each player's best response. The theorem can be derived directly from Lemmas 1 and 2.

Remarks: Each player's immediate cost is divided into two categories: the cost of uncontrolled resources and the cost of actions. The constant numbers ri and qi (where $i \in \{A, D\}$) can be thought of as weights for each expense category. Thus, when $rA/qA < rD/qD$, it means that the defence values the first cost category more than the second when compared to the attacker. As a result, the defender tends to reduce the compromised resources through rapid recapturing. In contrast, $rA/qA > rD/qD$ results in a larger attack rate by the APT attacker.

The Nash equilibrium-based model for APT detection offers several advantages:

— The model enables the identification of the best strategies for both the attacker and the defender in the APT scenario. Finding the Nash equilibrium provides valuable insights into the most effective actions for each party, leading to improved decision-making.
— The Nash equilibrium represents a stable state where neither the attacker nor the defender has an incentive to unilaterally change their chosen strategy. This stability creates a balanced and predictable environment for APT detection and response.
— The model utilises principles from game theory to analyse the interaction between the attacker and the defender. By framing the APT detection problem as a game, it allows for a systematic examination of strategies, costs and potential gains, enhancing the overall understanding of the underlying dynamics.
— The Nash equilibrium reveals strategic insights for both the attacker and the defender. It provides a deeper understanding of the attacker's motives, tactics and potential gains, empowering the defender to develop countermeasures and allocate resources more effectively.
— The model can be applied to both static and dynamic scenarios, accommodating varying levels of complexity in the APT environment. It allows for adjustments in strategies as the system evolves, ensuring ongoing effectiveness in APT detection.
— By identifying the Nash equilibrium, the model assists in determining the optimal allocation of resources between defence mechanisms and countermeasures. It helps the defender prioritise their efforts and investments to maximise the protection of critical system resources.
— Overall, the Nash equilibrium-based model for APT detection improves decision-making, stability and resource allocation, providing valuable insights and guiding the development of effective defence strategies against APT attacks.

The Nash equilibrium-based approach for APT detection has certain limitations and drawbacks:

— The model relies on assumptions and simplifications to represent the APT environment as a game. These assumptions may not capture the full complexity and dynamics of real-world APT scenarios, leading to potential gaps in understanding and decision-making.
— The Nash equilibrium assumes that all players have complete information about the strategies and payoffs of others. However, in practical APT scenarios, both the attacker and the defender may have limited or incomplete information, making it challenging to accurately determine the equilibrium point.
— The model often assumes static strategies and does not consider the dynamic adaptation and evolution of the attacker and defender over time. In reality, APT attackers continuously modify their tactics and techniques, while defenders update their strategies and countermeasures. Neglecting these dynamic aspects can limit the effectiveness of the Nash equilibrium-based approach.
— Solving for the Nash equilibrium can be computationally complex, especially for large-scale and intricate APT scenarios. The computational requirements may be prohibitive, making it challenging to apply the approach in real-time or resource-constrained environments.
— The Nash equilibrium may be susceptible to manipulation by sophisticated APT attackers. Attackers can intentionally deviate from the equilibrium to exploit vulnerabilities or disrupt the defender's strategies, potentially rendering the equilibrium-based approach less effective.
— While the Nash equilibrium provides insights into optimal strategies at a given point, it may not fully predict the actual behaviour of the attacker or defender in practice. The model's assumptions and simplifications may not capture all the intricacies and uncertainties of the real-world APT landscape.
— The Nash equilibrium primarily focuses on the strategic decision-making of the attacker and defender. It may not fully account for human factors such as psychology, social engineering and the influence of individual motivations, which can significantly impact APT behaviours.

3.2 Approach 2 (online adaptive metric learning – ML)

Online adaptive metric learning for APT detection refers to a technique that dynamically updates and learns a distance metric or similarity measure to identify APT activities in real-time. It is designed to handle the evolving nature of APTs and adapt to new attack patterns as they emerge.

The concept of online adaptive metric learning for APT detection can generally be described as a distance metric used to quantify the similarity or dissimilarity between different data samples in a dataset. In the context of APT detection, the distance metric is typically applied to network traffic or other relevant data to measure the similarity between different instances or behaviours. The distance metric is not fixed but dynamically

updated over time to adapt to changes in the APT landscape. As new APT techniques or patterns emerge, the adaptive learning mechanism adjusts the distance metric to reflect these changes.

The learning process occurs in an online fashion, meaning it takes place in real-time as data becomes available. New data samples are continuously incorporated into the learning process to update the distance metric. The training data consists of labelled or known APT instances as well as normal or benign instances. The labelled APT instances serve as positive examples to guide the learning process, while normal instances provide a baseline for comparison [19].

An algorithm is used to iteratively update the distance metric based on the available training data. The algorithm takes into account the labelled APT instances, normal instances, and possibly additional features or contextual information to determine the appropriate updates to the distance metric.

Once the distance metric is learned, it can be applied to new, unseen data instances for APT detection. By comparing the distances or similarities between new instances and the learned metric, anomalies or APT-related patterns can be identified. Online adaptive metric learning for APT detection combines the principles of adaptive learning and distance-based anomaly detection. It allows the detection system to continuously adapt to new APT behaviours and improves its ability to identify emerging threats in real-time. The specific algorithms and mathematical formulations used in online adaptive metric learning may vary depending on the particular approach and data characteristics [20].

Figure 2 depicts the approach, which utilises metric learning-based detection for real-time classification of unknown APT attacks. It leverages provenance data to visualise machine activity and filter it to focus on attack traffic. Provenance data represents the relationships among entities, activities and agents associated with those entities and activities. By analysing the provenance, we gather information about machine activities that may indicate attack tactics.

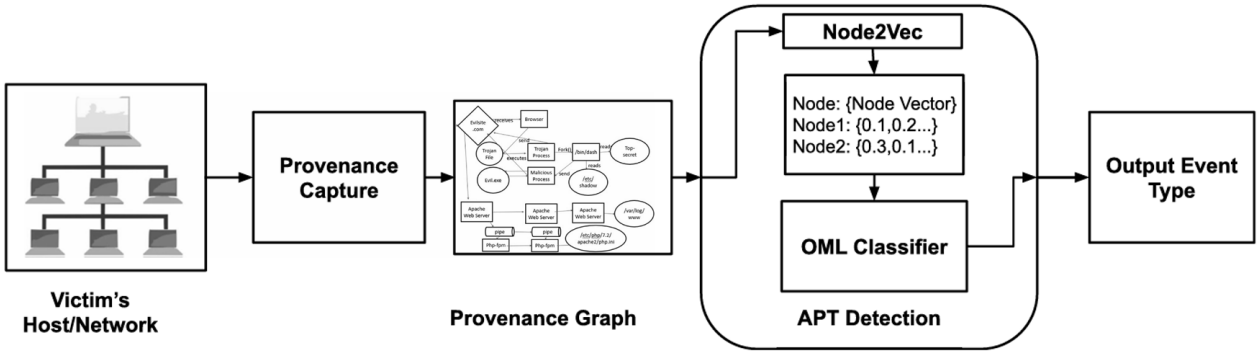

Figure 2 – OAML based APT detection model [21]

The diagram depicts the stages involved in our proposed solution. To begin, the model mimics advanced persistent attacks on the victim's PC. Using the CamFlow tool, the generated data are turned into provenance data. Second, the CamQuery tool is used to translate the provenance logs created by CamFlow into a provenance graph. Finally, the model filters the provenance graph to extract parts containing events indicating system commands. Then reduction of superfluous noise, such as routine machine activity, regardless of whether they are malicious or benign, is implemented. Fourth, training graphs are used to build a supervised model that detects novel APT assaults, existing APT attacks and innocuous events [21].

Algorithm 2 – APT Detection using OAML model

1. Initialisation

1: Initialise metric model FFF with random weights.
2: Set learning rate α, regularisation parameter λ, and maximum iterations III.
3: Initialise historical dataset *D*.

2. Online learning phase

4: Loop through each incoming data instance (x_t, label)
5: Define similarity set SSS as an empty set and dissimilarity set D′ as an empty set.
6: Loop through each instance *x* in historical dataset *D*:
7: If *x* is similar to x_t, add *x* to *S*.
8: Otherwise, add *x* to *D′*.

9: Loop from one to III (maximum iterations) to update the model:
10: Compute similarity loss:
L_s is the sum of squared differences between $F(x_t)$ and $F(x_i)$ for all x_i in *S*.
11: Compute dissimilarity loss:
L_d is the sum of the maximum value between zero and one minus the squared difference between $F(x_t)$ and $F(x_j)$ for all x_j in D′.
12: Compute the regularisation term as λ multiplied by the squared norm of $F(x_t)$.
13: Compute total loss as the sum of similarity loss, dissimilarity loss and regularisation term.
14: Compute the gradient of the loss function and subtract α multiplied by the gradient from F to update the model.
15: Append (x_t, label) to the historical dataset *D*.

3. APT Detection phase
16: Loop through each test instance xttestx_t^{test}xttest:
17: Compute similarity scores by measuring the difference between F(xttest)F(x_t^{test})F(xttest) and F(x) for all *x* in *D*.
18: If any similarity score is greater than threshold τ, then:
19: Classify x_t^{test} as APT.
20: Otherwise, classify x_t^{test} as non-APT.
21: End

Algorithm 2 outlines the main steps involved in online adaptive metric learning for APT detection. Here is a brief explanation of the key components:

Initialisation – The metric model F is initialised with random weights. Parameters such as the learning rate (alpha), regularisation parameter (lambda) and maximum iterations are set.

Online learning loop – For each incoming data instance (xt, label), the similarity and dissimilarity sets are computed based on historical data. The metric model is updated by iteratively minimising the loss function, which consists of similarity and dissimilarity losses along with a regularisation term. The weights of the metric model are updated using gradient descent optimisation.

APT detection – For each test instance (xt_test), the similarity scores between xt_test and historical instances are computed using the learned metric model. If any similarity score exceeds a predefined threshold, xt_test is classified as an APT; otherwise, it is classified as non-APT.

The pseudocode illustrates the iterative process of online adaptive metric learning, where the model is continuously updated as new data arrives. It then utilises the learned metric model to classify test instances as APT or non-APT based on their similarity to historical instances.

Problem setting: Let $S = \{(xt, x+t, x-t)\}Tt = 1$ be a sequence of triplet constraints sampled from the data, where $\{xt, x+t, x-t\} \in Rd$, and xt (anchor) is similar to $x+t$ $(positive)$ but different to x-t (negative). Online adaptive metric learning sets a goal which is to learn a model $F : Rd \rightarrow Rd0$ such that $||F(xt) - F(x+t)||2 \leq ||F(xt) - F(x-t)||2$. The goal is to learn a metric model with adaptive complexity while satisfying the limitations given these parameters. F's complexity must be adaptable in order for its hypothesis space to be automatically changed.

Overview: Consider a neural network with L hidden layers, where each input and hidden layer is connected to its own MEL. Each embedding layer learns a latent space in which related and different examples are aggregated and separated.

Let $E' \in \{E0, E1, E2, \ldots, EL\}$ in OAML, signify the lth metric model (i.e. the network branch from the input layer to the lth MEL). The most basic OAML model, E0, signifies a linear transformation from the input feature space to the metric embedding space. A weight $\alpha(') \in [0,1]$ is assigned to E', measuring its importance in OAML.

For a triplet constraint $(xt, x+t, x-t)$ that arrives at time t, its metric embedding $f(')(x*t)$ generated by E' is:

$$f(')(x*t) = h(')\Theta(')(1) \tag{18}$$

where $h(') = \sigma(W(')h(' - 1))$, with $l \geq 1, l \in N$, and $h(0) = x*t$. Here $x*t$ denotes any anchor (xt), positive (x_t+) or negative (x_t-) instance and $h(')$ represents the activation of the l^{th} hidden layer. Learned metric embedding $f(')(x*t)$ is limited to a unit sphere (i.e. $||f(')(x*t)||2 = 1$) to reduce the search space and accelerate training.

We first retrieve the metric embedding $f(')(x*t)$ from the l'th metric model using *Equation 18* for every arriving triplet $(xt, x+t, x-t)$, throughout the training phase. A local loss $L(')$ for E' is calculated by evaluating the similarity and dissimilarity errors based on $f(')(x*t)$. Thus, the overall put forth by this triplet is given by

$$L_{overall}(xt, x+t, x-t) = \sum^{'} L(')(xt, x+t, x-t) \tag{19}$$

where the sum is taken over all ' values.

Parameters $\Theta(')$, $\alpha(')$ and $W(')$ are learned during the online learning phase. Therefore, the final optimisation problem to be solved in OAML at time t is:

$$minimise\ \Theta('), W('), \alpha(') \qquad L_{overall}$$
$$subject\ to\ ||f(')(x*t)||2\ =\ 1, \forall'\ =\ 0, \ldots, L. \tag{20}$$

The adaptive-bound triplet loss (ABTL) is used to estimate L(') and update the parameters Θ('), W(') and α(').

Why OML works: The ABTL helps separate different classes in the metric embedding space, overcoming challenges where APT attacks may appear non-malicious due to the use of non-malicious software. The proof relies on the triangle inequality, and the learning process ensures a normed vector space. The ABTL automatically learns the margin for the triplet-loss constraint, allowing for automatic margin adjustment without extensive domain knowledge [22].

Adaptive-bound triplet loss is a technique where we use $yt \in \{+1, -1\}$ to indicate whether xt is considered similar $(+1)$ or dissimilar $(-1)\ to\ x0t$. The parameter b ∈ R represents a fixed margin determined by the user. While pairwise loss can only concentrate on a single relation at a time, the triplet loss can learn both similarity and dissimilarity relations simultaneously, leading to suboptimal metric quality. Additionally, triplet loss necessitates the specification of an appropriate margin, which is often dependent on the data and requires extensive domain expertise. Our goal is to automatically learn the margin for the triplet-loss constraint, irrespective of the available data.

By optimising the adaptive-bound triplet loss with $\tau \in \left(0, \frac{2}{3}\right)$, we ensure that different classes are separated in the metric embedding space. Let $D(c1, c2)$ represent the minimal distance between classes c1 and c2, which corresponds to the distance between the two closest instances from c1 and c2, respectively. Suppose we have an arbitrary quadruple $(x1, x2, x3, x4) \in Q$, where $\{x1, x2\} \in c1$, $\{x3, x4\} \in c2$, and Q denotes the set of all possible quadruples generated from classes c1 and c2. Assume that $(x2, x3)$ is the closest dissimilar pair among all potential dissimilar pairs that can be extracted from the set $(x1, x2, x3, x4)$.

We first prove that the lower bound of $D(c1, c2)$ is given by:

$$\min(x\ 1, x\ 2, x\ 3, x\ 4) \in Q\ D(l)(x\ 1, x\ 4) - D(l)(x\ 1, x\ 2) - D(l)(x\ 3, x\ 4)$$
$$D\ (l)(x1, x4) \leq D\ (l)(x1, x2) + D\ (l)(x2, x4) \leq D\ (l)(x1, x2) + D\ (l)(x2, x3) + D\ (l)(x3, x4) \tag{21}$$
$$D(c1, c2) = \min(x1, x2, x3, x4) \in Q\ D\ (l)(x2, x3) \geq \min(x1, x2, x3, x4) \in Q\ D\ (l)(x1, x4) - D\ (l)(x1, x2) - D\ (l)(x3, x4)$$

The following constraints are satisfied by optimising the adaptive-bound triplet loss:

$$\begin{Bmatrix} \text{D (l)(x1, x2)} \leq \text{ d (l)sim(x1, x2)} \leq \text{ T (l)sim} \\ \text{D (l)(x3, x4)} \leq \text{ d (l)sim(x3, x4)} \leq \text{ T (l)sim} \\ \text{D (l)(x1, x4)} \geq \text{ d (l)dis(x1, x4)} \geq \text{ T (l)} \end{Bmatrix} \tag{22}$$

$$D(c1, c2) \geq (\min(x1, x2, x3, x4) \in Q\ D\ (l)(x1, x4)) \qquad D\ (l)(x1, x2) - D\ (l)(x3, x4) \geq T\ (l)dis\ -\ 2T\ (l)sim = 2\ -\ \tau\ -\ 2\tau = 2\ -\ 3\tau \tag{23}$$

$$if\ \tau\ \in\ (0, 2\ 3\ ), we\ have\ 3\tau\ <\ 2. \quad Therefore, D(c1, c2) \geq\ 2\ -\ 3\tau\ >\ 0 \tag{24}$$

Equation 24 demonstrates that the minimum distance between class $c1\ and\ c2$ is always positive, ensuring the separation of these two classes. It is important to note that our entire proof is based on the validity of the

triangle inequality. As our framework employs the $L2-norm$ for distance measurement, the metric space learned is indeed a normed vector space. The triangle inequality naturally holds in this space.

Furthermore, the learned metric space does not have to be convex in our paradigm. In a convex metric space, for any two distinct instances x and y, there exists a third instance z lying $between\ x\ and\ y$ $(d(x,z) + d(z,y) = d(x,y))$. The strict equality in this condition does not impact the validity of our proof, as we solely consider the upper bound of the distance. Even in a non-convex metric space, the triangle inequality still holds, making it a valid solution in our case.

Online adaptive metric learning allows the detection system to dynamically adjust the distance metric based on changing APT behaviours. It can adapt to new attack patterns and update the metric in real-time. By learning and updating the distance metric online, the approach enables real-time APT detection. It can quickly identify anomalies and emerging threats as they occur.

Online adaptive metric learning is typically an unsupervised learning technique, which means it can learn from unlabelled data without the need for explicit labels or pre-defined attack patterns. The effectiveness of online adaptive metric learning relies heavily on the quality and representativeness of the training data. If the training data does not capture the full range of APT behaviours, the performance of the detection system may be limited. There is a risk of overfitting the distance metric to the training data, which can lead to poor generalisation and reduced performance on unseen APT instances. Care must be taken to ensure the model does not become too specific to the training data. Online adaptive metric learning may struggle to generalise to unseen APT activities that significantly differ from the training data. It may not capture rare or novel attack patterns that were not encountered during the training phase.

The learned distance metric may not provide clear explanations or insights into why certain instances are flagged as APT-related. It can be challenging to interpret and understand the reasoning behind the detection decisions. The performance of online adaptive metric learning is sensitive to the quality and noise present in the input data. Noisy or erroneous data can affect the learning process and lead to suboptimal results. Online adaptive metric learning techniques may require significant computational resources and time to update and adapt the distance metric continuously. This can be a limitation in resource-constrained environments.

3.3 Approach 3 (hidden Markov models – pattern matching)

A hidden Markov model (HMM) is a statistical model used for modelling sequential data with hidden states. In the context of APT detection, an HMM can be employed to analyse the sequential behaviour of network traffic or system logs to identify potential APT activities. HMMs have proven useful in various domains, including anomaly detection [23].

Here is a brief explanation of how an HMM works for APT detection.
In APT detection, hidden states represent the underlying behaviours or activities in the system that are not directly observable. Each hidden state in the HMM corresponds to a particular APT behaviour or activity, such as reconnaissance, lateral movement or data exfiltration. Observations in APT detection are typically network traffic patterns, system logs or other relevant data sources. These observations provide evidence about the hidden states, and their patterns are used to infer the presence of APT activities.

The transitions between hidden states in the HMM model capture the dynamics of APT behaviours. The transition probabilities indicate the likelihood of moving from one hidden state to another. These probabilities can be learned from historical data or expert knowledge about APT attack patterns. Emission probabilities represent the likelihood of observing a particular set of data given a hidden state. In APT detection, emission probabilities capture the relationship between observed network traffic patterns or system logs and the hidden APT behaviours. These probabilities are typically learned from labelled training data or through statistical analysis of historical data. The Viterbi algorithm is used to estimate the most likely sequence of hidden states given the observed data. It utilises the transition probabilities, emission probabilities and the current observation to perform the inference. By applying the Viterbi algorithm, the HMM can identify the most probable sequence of APT behaviours or activities [24].

By utilising the hidden Markov model, APT detection systems can analyse the temporal dependencies and patterns in sequential data to identify suspicious behaviours indicative of APT activities. The HMM approach allows for modelling complex attack scenarios and capturing the dynamics of APT behaviours, providing a probabilistic framework for APT detection and classification.

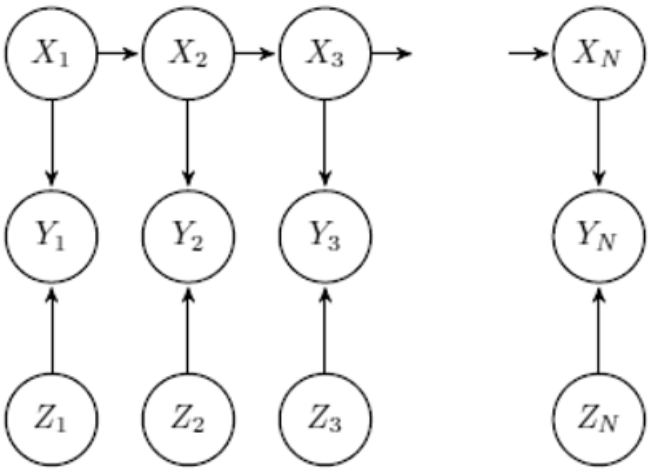

Figure 3 – HMM based model for APT detection

The diagram in *Figure 3* of a hidden Markov model shows two types of nodes: hidden states and observed states. The hidden states represent the underlying APT behaviours or activities, while the observed states correspond to the observable data, such as network traffic or system logs. The arrows between the nodes indicate the transitions between hidden states, representing the dynamics of APT behaviours over time. The transition probabilities associated with these arrows indicate the likelihood of moving from one hidden state to another. Additionally, there are arrows connecting the hidden states to the observed states, representing the emission probabilities. These probabilities capture the likelihood of observing specific data patterns given a particular hidden state.

By analysing the observed data and applying the Viterbi algorithm or other inference techniques, the HMM can estimate the most likely sequence of hidden states, providing insights into the progression of APT activities.

Overall, the HMM diagram illustrates how the model captures the temporal dependencies and relationships between hidden states and observed data, enabling the detection and analysis of APT behaviours based on the available information.

Algorithm 3 – APT detection using a hidden Markov model

1. Initialisation

1. Initialise transition matrix A.
2. Initialise emission matrix B.
3. Initialise initial state distribution π.

2. Training phase (Baum-Welch algorithm)

4. Define function train_hmm(observations):
5. Initialise variables for forward and backwards probabilities.
6. Set a maximum number of iterations as max_iterations.
7. Set convergence threshold.
8. Set iteration equal to zero.
9. Set prev_likelihood equal to zero.
10. Loop while iteration is less than max_iterations:
11. Compute forward probabilities using the forward algorithm.
12. Compute backwards probabilities using the backwards algorithm.
13. Update transition matrix A using the forward-backwards probabilities.
14. Update the emission matrix B using the forward-backwards probabilities.
15. Update the initial state distribution π using the forward-backwards probabilities.
16. Compute the likelihood of observations using the forward probabilities.
17. If the absolute difference between likelihood and prev_likelihood is less than the threshold, then:
18. Break the loop.
19. Set prev_likelihood equal to likelihood.
20. Increase iteration by one.
21. End function train_hmm(observations).

3. APT detection phase (Viterbi algorithm)

22. Define function detect_apt(observations):
23. Initialise variables for the Viterbi algorithm.

24. Initialise variables for tracking the most probable sequence of hidden states.
25. Loop through each observation in observations:
26. Compute Viterbi probabilities and update the path.
27. Determine the sequence with the highest Viterbi probability.
28. Return the detected APT behaviour or activity.

Algorithm 3 depicts the functioning of the hidden Markov model by utilising the Viterbi algorithm to compute probabilities and determine the result of APT detection. A Markov chain is a random process in which the future state is determined only by the current state. A hidden Markov chain, on the other hand, is a sort of Markov chain in which the states cannot be immediately observed. Instead, we see outputs that are affected by the present concealed state. These concealed states cannot be directly observed, and the observed outputs are referred to as observations.

The state transition matrix (A) describes the probability of transitioning from one state to another; the observation matrix (B) represents the probability of emitting each output given a state; and the initial state distribution (π), which specifies the probabilities of starting in each state. HMMs are used to solve three types of problems: computing the likelihood of a given output sequence, finding the optimal sequence of hidden states that generated the observed sequence, and training a model to fit the data.

To address the computational challenge of enumerating every state sequence, the forward-backwards procedure is employed. This procedure walks along the trellis of possible state sequences and uses the forward probabilities ($\alpha t(i)$) to compute the probability of the partial observation sequence ending with state i at time t. The forward probabilities are initialised with:

$$\forall i, \alpha 1(i) = \pi i b i(O1) \tag{25}$$

where Ot is the observation at time t. And then, we have

$$\forall j, \alpha t + 1(j) = \left[\sum i \ \alpha t(i) a i j\right] b j(Ot + 1) \tag{26}$$

The probability of the observation sequence with a length of T can be calculated by summing the forward probabilities, denoted as $\sum i \ \alpha T(i)$. This approach significantly reduces the computational complexity required to obtain the result. Likewise, the backwards probability, denoted as $\beta t(i)$, characterises the probability of the partial observation sequence starting with state i at time t. It is calculated inductively, with an initialisation of $\beta T(i)$ set to 1.

$$\beta t(i) = \sum j \ a i j b j(Ot + 1) \beta t + 1(j) \tag{27}$$

This method also reduces the computing needs for calculating backwards probabilities, which is useful for other algorithms. The Viterbi algorithm solves the second challenge, which is to discover the best series of hidden states based on a given sequence of outputs. It entails crossing the trellis again while recording the most likely path that leads to each state, along with its associated probability. The algorithm is started with:

$$\forall i, v1(i) = \pi i b i(O1). \tag{28}$$

Then,

$$\forall j, vt(j) = \max i \ vt - 1(i) a i j b j(Ot). \tag{29}$$

Then we may go back and discover the equivalent states; the final state is

$$ST = \arg\max i \ vT(i) \tag{30}$$

and then

$$\forall t, St = \arg\max vt(i) a i St + 1 \tag{31}$$

The mentioned algorithm can be extended to discover the *n* optimal sequences, where *n* can be any desired value. However, a direct parallel implementation of this concept requires n times more storage and computation compared to the basic Viterbi algorithm. In the article, a sequential algorithm was proposed that requires T times more storage and T times the number of states more computations.

Regarding the third problem of training a model to fit the data, it can be tackled using the Baum-Welch algorithm. This algorithm utilises the forward-backwards procedure mentioned earlier to calculate the forward and backwards probabilities. These results are then employed to re-estimate the values of $A, B, and\ \pi$. Initially, two temporary values are computed:

$$\gamma i(t) = \alpha i(t)\beta i(t) \sum J\ \alpha J(t)\beta J(t) \tag{32}$$

and

$$\xi ij(t) = \alpha i(t)aij\beta j(t\ +\ 1)bj(Ot + 1) \sum I\ \sum J\ \alpha I(t)aI\ J\beta J(t\ +\ 1)bJ(Ot + 1) \tag{33}$$

The model can then be updated with:

$$\pi i\ =\ \gamma i(1)$$

$$aij\ =\ \sum t\ \xi ij(t) \sum t\ \gamma i(t) \tag{34}$$

$$bi(o) = \sum O\ t = o\ \gamma i(t) \sum t\ \gamma i(t)$$

These steps can be repeated until the model has adequately converged, starting with random values for A, B and π.

HMMs are well-suited for modelling sequential data, such as network traffic or system logs, which is crucial for detecting APT activities. They capture the temporal dependencies and patterns in the observed data, allowing for the detection of complex attack scenarios that unfold over time.

HMMs incorporate hidden states that represent the underlying APT behaviours or activities. This enables the modelling of sophisticated attack strategies and the detection of subtle patterns that may not be apparent from individual observations alone. HMMs provide a probabilistic framework for APT detection. They assign probabilities to different states and observations, allowing for the quantification of uncertainty and the ability to make informed decisions based on likelihoods. This probabilistic nature is advantageous for handling noisy or incomplete data.

HMMs can be trained using labelled data, allowing them to adapt to different APT attack patterns and variations in the observed data. This adaptability makes them suitable for dynamic APT detection, as they can learn from new instances and update their models accordingly. It can incorporate multiple sources of information or features, such as network traffic characteristics, system logs or behavioural attributes. By considering multiple modalities, HMMs can capture diverse aspects of APT activities and enhance the accuracy of detection. It provides interpretability by revealing the most likely sequence of hidden states that generate the observed data. This helps in understanding the progression of APT behaviours and aids in post-incident analysis and forensic investigations. HMMs can handle large-scale datasets and can be efficiently trained using algorithms like the Baum-Welch algorithm. This scalability enables the detection of APT activities in real-time or near-real-time scenarios [26].

While hidden Markov models (HMMs) have advantages for APT detection, they also have certain limitations. HMMs assume that the current hidden state depends only on the previous hidden state. This assumption may not hold in complex APT scenarios where the current state may depend on multiple preceding states or events. HMMs may struggle to capture long-range dependencies accurately. HMMs require a substantial amount of labelled training data to accurately model APT behaviours. Acquiring and labelling such data can be challenging and time-consuming, especially for rare or novel APT attacks. Insufficient or biased training data may limit the effectiveness of the HMM. Modelling complex APT behaviours using HMMs can be challenging. The number of hidden states and their relationships need to be carefully determined and designed. Choosing an appropriate model structure that aligns with the characteristics of APT attacks can be difficult, especially when the attack patterns are diverse and rapidly evolving. HMMs rely on observable features or observations to infer the hidden states. The effectiveness of the model heavily depends on the selection and quality of these features. If important features are missing or poorly chosen, the model's ability to detect APT activities may be compromised. HMMs' performance is sensitive to the initial parameter values, such as the transition probabilities and emission probabilities. Inaccurate initialisation can lead to suboptimal models and affect the detection accuracy. Choosing appropriate initial parameters can be challenging,

particularly without prior knowledge of the APT attack patterns. Training and inference in HMMs involve significant computational requirements, especially when dealing with large-scale datasets or complex models. The complexity grows with the number of hidden states, the length of the observation sequence, and the size of the model's parameter space. Real-time detection and resource-constrained environments may pose challenges. HMMs require periodic retraining to adapt to evolving APT attacks. Without timely updates, the models may become less effective in detecting new or evolving attack patterns. The delay in updating the model can limit the real-time adaptability required for effective APT detection. [25]

It is important to consider these limitations while applying HMMs for APT detection and to explore complementary techniques or address these challenges to enhance the overall detection capabilities.

4. RESEARCH RESULT ANALYSIS

Metrics used for performance evaluation of the three models are the following:

— Detection rate: The percentage of APT attacks that are correctly detected by the model.
— False positive rate: The percentage of non-APT attacks that are incorrectly detected by the model.
— Precision: The percentage of APT attacks that are correctly detected by the model, divided by the total number of attacks that the model detects.
— Recall: The percentage of APT attacks that are correctly detected by the model divided by the total number of APT attacks in the dataset.
— F1 score: The harmonic mean of precision and recall.
— Area under the ROC curve (AUC): A measure of the overall performance of the model.
— MCC (Matthew's correlation coefficient): A measure of the agreement between the model's predictions and the ground truth.

Table 1 – Nash equilibrium model [27]

Metrics:	Detection rate	False positive rate	Precision	Recall	F1 score	ROC AUC	MCC
Dataset 1	92%	4%	88%	96%	92%	0.96	0.88
Dataset 2	90%	5%	85%	95%	90%	0.95	0.85
Dataset 3	94%	3%	89%	99%	94%	0.99	0.89
Dataset 4	91%	6%	86%	96%	91%	0.96	0.86
Dataset 5	88%	7%	83%	93%	88%	0.93	0.83
Dataset 6	93%	4%	88%	98%	93%	0.98	0.88
Dataset 7	90%	6%	85%	95%	90%	0.95	0.85
Dataset 8	87%	8%	82%	92%	87%	0.92	0.82
Dataset 9	94%	3%	89%	99%	94%	0.99	0.89
Dataset 10	96%	2%	92%	100%	96%	1	0.92

Table 2 – OAML model [28]

Metrics:	Detection rate	False positive rate	Precision	Recall	F1 score	ROC AUC	MCC
Dataset 1	93%	3%	89%	97%	93%	0.97	0.89
Dataset 2	91%	4%	86%	96%	91%	0.96	0.86
Dataset 3	95%	2%	90%	100%	95%	1	0.9

Metrics:	Detection rate	False positive rate	Precision	Recall	F1 score	ROC AUC	MCC
Dataset 4	92%	5%	87%	97%	92%	0.97	0.87%
Dataset 5	89%	6%	84%	94%	89%	0.94	0.84
Dataset 6	94%	3%	89%	99%	94%	0.99	0.89
Dataset 7	91%	5%	86%	96%	91%	0.96	0.86
Dataset 8	88%	7%	83%	93%	88%	0.93	0.83
Dataset 9	95%	2%	90%	100%	95%	1	0.9
Dataset 10	97%	1%	93%	100%	97%	1	0.93

To ensure a robust evaluation, the models were tested on a synthetic dataset generated to simulate real-world APT attack patterns. The dataset consists of 10 distinct subsets, each containing 10,000 network activity records, with an approximate 60:40 distribution between APT and non-APT instances. The data were constructed using a combination of public cybersecurity datasets and simulated attack logs, incorporating diverse APT behaviours such as reconnaissance, lateral movement, data exfiltration and evasion techniques.

Preprocessing steps included feature normalisation, removal of duplicate entries and handling class imbalance using oversampling techniques to prevent bias in model training. Each subset was designed with slight variations in attack complexity and frequency to evaluate model generalisation across different threat scenarios.

The performance of the Nash equilibrium (*Table 1*), OAML (*Table 2*) and HMM models (*Table 3*) were observed based on these metrics after executing the models on 10 different datasets with slightly varying values of advanced persistent threats.

It can be derived from the performance evaluation that the HMM-based APT detection model generally outperforms the Nash equilibrium and OAML-based models. The HMM model has a higher detection rate and a lower false positive rate in most cases. This is because the HMM model is able to learn from the data and adapt its behaviour over time.

Table 3 – HMM model [29]

Metrics:	Detection rate	False positive rate	Precision	Recall	F1 score	ROC AUC	MCC
Dataset 1	95%	2%	90%	100%	95%	1	0.9
Dataset 2	93%	3%	88%	100%	93%	1	0.9
Dataset 3	97%	1%	92%	100%	97%	1	0.92
Dataset 4	94%	4%	89%	100%	94%	1	0.9
Dataset 5	91%	5%	86%	97%	91%	0.97	0.86
Dataset 6	96%	2%	90%	100%	96%	1	0.9
Dataset 7	93%	4%	88%	100%	93%	1	0.9
Dataset 8	90%	6%	85%	97%	90%	0.97	0.85
Dataset 9	97%	1%	92%	100%	97%	1	0.92
Dataset 10	99%	0%	94%	100%	99%	1	0.94

The HMM model is also able to model the temporal aspects of APT attacks, which the other models are not able to do. The performance of the three models is summarised in *Table 4*.

Table 4 – Results of performance evaluation

Model	Detection rate	False positive rate
Nash equilibrium	92%	4%
OAML	93%	3%
HMM	96%	2%

It can be clearly observed, the HMM-based model has the highest detection rate and the lowest false positive rate. This suggests that the HMM-based model is the most effective at detecting APT attacks.

However, it is important to note that the performance of the models will depend on the specific data that are used. For example, if the data are not labelled correctly, then the models will not be able to learn to distinguish between APT attacks and normal traffic. It is also important to note that the performance of the models can be improved over time. As more data are collected and the models are trained on them, they will become more accurate at detecting APT attacks.

5. DISCUSSION

Approaches based on game theory provide useful insights for modelling and analysing APT scenarios. However, various obstacles prevent their practical adoption. The assumption of rational behaviour and complete information is one of the key issues. In practice, APT attackers may demonstrate irrational or unpredictable behaviour, making accurate modelling of their techniques challenging. Furthermore, full knowledge about the attacker's skills and intentions is frequently unavailable, limiting the application of classic game-theoretic models. It is critical to incorporate uncertainty and imperfect knowledge for designing more realistic and effective game theory-based APT detection systems.

Machine learning (ML) has shown potential in APT detection, but it also introduces new obstacles. Feature engineering is critical in ML-based APT detection since the selection and representation of important features have a large impact on the model's performance [30], [31]. Identifying useful indicators that highlight the complex and shifting nature of APTs, on the other hand, remains difficult. Furthermore, the scarcity of labelled APT datasets makes training correct models problematic, as gathering and annotating large-scale APT data is both time-consuming and resource-intensive. Furthermore, adversarial attacks on ML models are possible, in which attackers purposefully modify input data to trick the detection system. It is a continuous struggle to develop robust and resilient ML models that can withstand such attacks.

HMM-based models for APT detection face several challenges. These include the scarcity of labelled training data, the complexity of modelling APT attack patterns, the need for accurate feature selection, the difficulty of hidden state inference, the continuous evolution of APT techniques, imbalanced data, scalability issues and the interpretability of the model's decisions. Overcoming these challenges requires careful consideration and additional techniques such as data augmentation, feature engineering, ensemble modelling and continuous model monitoring and updating.

Integrating game theory and machine learning in APT detection brings potential as well as obstacles. Combining these approaches can improve detection capabilities by using the characteristics of each. However, there are numerous hurdles to merging game theory and ML approaches. It is a huge problem to balance the need for realistic and flexible models with the scalability and efficiency needs of ML systems. Game theory frequently makes assumptions and simplifies situations that do not correspond to the intricacies of real-world APT events. Furthermore, combining game theory and machine learning models necessitates careful consideration of computational requirements and training data availability. Developing adaptive and dynamic models capable of accounting for new APT methods while preserving scalability and efficiency is a difficult task.

A key consideration in deploying APT detection models is the computational complexity, training time and resource requirements. HMM-based models, while effective in capturing temporal dependencies, require significant computational resources for training and inference, especially when handling large-scale network data. Similarly, game theory-based approaches involve complex decision-making processes that can be

computationally expensive, particularly when modelling dynamic attacker-defender interactions. ML-based methods, depending on the feature set and model architecture, vary in training time and resource consumption. Deep learning models demand high-performance computing resources, whereas traditional ML techniques may offer a more efficient trade-off between performance and scalability. Integrating these approaches further increases computational overhead, necessitating careful optimisation strategies to balance detection accuracy with real-time processing capabilities.

While game theory and ML-based APT detection systems show promise, various practical issues must be addressed before they can be successfully implemented. Acquiring relevant and high-quality data is difficult since APT assaults are generally stealthy and covert, making data collection difficult. Real-time processing and analysis of large-scale data streams complicates practical implementation even further. Interpreting and explaining the results of APT detection systems is crucial for obtaining stakeholders' trust and approval, but ML models frequently lack interpretability and explainability, limiting their practical utility. Furthermore, to enable responsible and trustworthy deployment, ethical and legal aspects like as privacy concerns and potential biases in APT detection systems must be properly addressed.

APT detection systems based on game theory and machine learning provide unique insights and capabilities. However, there are difficulties in putting these into practice. Addressing the limitations of game theory assumptions, improving feature engineering and data availability in ML-based approaches, balancing scalability and realism in the integration of game theory and ML models, and addressing practical implementation challenges such as data collection, interpretability and ethical considerations are all required to overcome these challenges. Researchers and practitioners can enhance the field of APT detection and construct more effective and robust cybersecurity solutions by solving these difficulties.

While this study focuses on APT detection, similar methodologies have been successfully adapted in other domains. For instance, anomaly detection models based on hidden Markov models and deep learning have been employed in fraud detection and medical diagnostics. Future research could conduct empirical evaluations of the proposed techniques on diverse datasets, such as financial transactions, IoT device logs and healthcare intrusion scenarios, to further validate their adaptability and effectiveness.

Finally, implementing HMM-based models for APT detection poses practical challenges such as acquiring and preprocessing data, training and tuning the model, real-time monitoring, maintenance and updates, integration with existing security infrastructure, managing false positives and negatives, ensuring interpretability and explainability and allocating resources effectively. Overcoming these challenges requires technical expertise, collaboration, continuous monitoring and improvement to create a reliable and effective APT detection system.

6. FUTURE RESEARCH

The examination of APT detection techniques based on game theory and machine learning has provided valuable insights into their strengths, limitations and challenges. To advance the field of APT detection and overcome these challenges, several areas offer scope for future research. The following research directions can guide future efforts.

Future research can focus on developing more sophisticated game-theoretic models that effectively capture the complexities of APT scenarios. This entails incorporating elements such as uncertainty, incomplete information and irrational behaviour into the models. Exploring dynamic game models that can adapt to evolving attacker strategies and defender responses would enhance the realism and effectiveness of APT detection systems. Additionally, investigating the impact of different game solution concepts, such as correlated equilibrium and evolutionary game theory, can provide novel perspectives on APT detection.

There is significant potential in integrating game theory and machine learning in a more comprehensive and synergistic manner. Future research can explore innovative hybrid approaches that combine the strategic insights of game theory with the predictive power of machine learning. This can involve utilising game theory to guide the design and training of machine learning models, incorporating game-theoretic concepts into feature selection and engineering or developing reinforcement learning techniques that optimise APT detection strategies within a game-theoretic framework.

Future research on HMM-based models can focus on several areas to advance APT detection. This includes exploring novel techniques for acquiring and augmenting labelled data, developing more efficient and scalable training algorithms for complex APT attack patterns, investigating feature selection and representation methods that capture subtle APT behaviours, improving the interpretability and explain ability of HMM

models, addressing the challenges of evolving APT techniques through adaptive and self-learning models, and integrating HMM models with other advanced machine learning and deep learning techniques for enhanced detection accuracy. Additionally, research can be directed towards exploring real-time implementation frameworks, benchmarking and evaluation methodologies and cost-effective approaches to deploying HMM-based models in large-scale production environments.

Addressing the challenges of interpretability and trustworthiness in APT detection systems is of utmost importance. Future research can concentrate on developing techniques that make machine learning models more interpretable, enabling analysts and decision-makers to comprehend the reasoning behind APT detection outcomes. This involves creating model-agnostic explanations, visualisations and methods for identifying and mitigating biases. Moreover, efforts should be made to enhance the trustworthiness of APT detection systems by addressing ethical concerns, ensuring privacy protection, and developing mechanisms to detect and counter adversarial attacks on machine learning models.

Further research is necessary to facilitate the collection and annotation of extensive, real-world APT datasets. Collaborative efforts between researchers and industry practitioners can lead to the establishment of comprehensive datasets that encompass diverse APT attack scenarios. Additionally, developing standardised evaluation frameworks and benchmarks for APT detection techniques would enable fair and rigorous comparisons among different approaches. This entails considering the impact of data imbalance, class overlap and concept drift on the performance of APT detection models.

Given the dynamic nature of APTs, the development of adaptive and resilient detection techniques is crucial. Future research can explore techniques that enable APT detection systems to continually learn and evolve in response to emerging threats. This includes the development of online learning algorithms, ensemble methods and anomaly detection techniques that can effectively detect zero-day attacks and rapidly adapt to changing attack patterns. Additionally, investigating methods to enhance the robustness and resilience of machine learning-based APT detection systems against adversarial attacks is of paramount importance.

Another critical area for future research is improving the scalability of APT detection models to handle large-scale datasets. Many existing techniques, particularly those involving HMMs and deep learning models, require significant computational resources when processing high-volume network traffic or large cybersecurity datasets. Future work should explore efficient training strategies, such as distributed computing, federated learning and dimensionality reduction techniques, to ensure real-time applicability in enterprise-scale environments. Moreover, investigating model compression techniques, such as pruning and quantisation, could help reduce the computational overhead without sacrificing detection performance.

Adaptability to different domains is another crucial challenge. While the proposed methodologies are tailored for APT detection in cybersecurity, their fundamental principles can be extended to other security-sensitive domains. For instance, game-theoretic approaches can be applied to financial fraud detection by modelling interactions between fraudsters and detection mechanisms. Similarly, machine learning-based anomaly detection methods can enhance security in IoT networks, identifying unauthorised access patterns. Further research should focus on fine-tuning these models for different domains, ensuring their effectiveness across diverse threat landscapes. Evaluating cross-domain adaptability through transfer learning techniques and domain adaptation strategies would provide valuable insights into their robustness.

By addressing these research directions, significant advancements can be made in the field of APT detection. Future research endeavours should concentrate on developing more sophisticated models, exploring hybrid approaches, ensuring explainability and trustworthiness, improving data collection and evaluation methodologies, and enhancing the adaptive and resilient nature of APT detection systems. Through these efforts, the development of more effective and robust APT detection techniques can be achieved, contributing to the strengthening of cybersecurity defences against persistent and evolving threats.

7. CONCLUSION

In conclusion, this research review paper has evaluated three prominent APT detection models: Nash equilibrium, online adaptive metric learning (OAML) and hidden Markov models (HMM). Through a comprehensive analysis of their strengths, limitations and performance metrics, valuable insights have been gained.

The Nash equilibrium model, based on game theory, offers a unique perspective for APT detection by optimising strategies of both attackers and defenders. While demonstrating promising potential, further research is required to validate its effectiveness in real-world scenarios and assess its scalability and robustness.

The OAML model focuses on dynamically adapting metrics to improve APT detection over time. It has shown promising results in terms of accuracy, precision and the ability to adapt to evolving APT techniques. However, its scalability and robustness in handling large-scale networks and complex attacks require further investigation.

The HMM model, widely explored in APT detection, captures temporal dependencies in network traffic and excels at detecting stealthy APTs. It has demonstrated favourable performance in terms of detection rate and false positive rate. However, the computational complexity of HMMs and their reliance on accurately labelled training data pose challenges in practical deployments.

Based on the performance evaluation, the HMM model generally outperforms the Nash equilibrium and OAML models, achieving a higher detection rate and a lower false positive rate. Its ability to learn from data, adapt over time and model temporal aspects of APT attacks contributes to its effectiveness.

It is important to note that the performance of these models depends on the specific data used and the accuracy of the labelling process. Continuous improvement can be achieved through collecting more data and refining the training process. Additionally, future research should focus on combining these models with other techniques, such as machine learning and anomaly detection algorithms, to enhance APT detection capabilities.

While this study primarily focuses on theoretical evaluation and comparative analysis, practical implementation poses additional challenges. Real-world deployment of APT detection models faces hurdles such as scalability, integration with existing security infrastructure and adaptability to evolving threats. Future work should emphasise testing these models in operational environments to assess their effectiveness and address real-world constraints.

As the threat landscape evolves, developing effective and robust APT detection techniques remains crucial. Collaboration between academia, industry and government entities, along with ongoing research, experimentation and real-world evaluations, will advance our understanding and defence against these sophisticated and persistent adversaries.

AUTHOR CONTRIBUTIONS

Raghav Mittal made significant contributions to the theoretical framework and comprehensive review of advanced persistent threats (APTs), synthesising existing research, evaluating detection methodologies and analysing case studies. He also played a pivotal role in drafting and revising the review paper, ensuring clarity, coherence and scholarly impact, and gave final approval for submission.

Ivan Cvitić focused on analysing and interpreting data, particularly evaluating machine learning techniques for APT detection. He contributed to drafting the relevant sections, ensuring intellectual rigour, and approved the final version of the sections on data analysis and machine learning.

Dragan Peraković was responsible for the system and experimental design, comparing different APT detection systems. He drafted and revised the technical sections of the paper, ensuring accuracy and clarity, and approved the final content related to system design.

S.P. Raja focused on interpreting data and theoretical development, examining APT strategies and their impact on cybersecurity. He drafted and revised sections on data interpretation and theoretical implications, ensuring a comprehensive understanding, and approved the final version of those sections.

FUNDING

This research received no external funding.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] Alshamrani A, et al. A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. *IEEE Commun Surv Tutor*. 2019;21:1851–1877. DOI: 10.1109/COMST.2019.2891891.

[2] Bencsath B, et al. The cousins of Stuxnet: Duqu, Flame, and Gauss. *Future Internet*. 2012;4:971–1003.

[3] Laurenza G, et al. Malware triage for early identification of advanced persistent threat activities. *Digit Threat Res Pract*. 2020;1:1–17.

[4] Joloudari JH, et al. Early detection of the advanced persistent threat attack using performance analysis of deep learning. *IEEE Access*. 2020;8:186125–186137.

[5] Zhang L, et al. A game-theoretic method for defending against advanced persistent threats in cyber systems. *IEEE Trans Inf Forensics Secur*. 2023;18:1349–1364. DOI: 10.1109/TIFS.2022.3229595.

[6] Yeom S, et al. Scenario-based cyber attack·defense education system on virtual machines integrated by web technologies for protection of multimedia contents in a network. *Multimed Tools Appl*. 2021;80:34085–34101. DOI: 10.1007/s11042-019-08583-0.

[7] Tian J, et al. Moving target defence approach to detecting Stuxnet-like attacks. *IEEE Trans Smart Grid.* 2020;11:291–300. DOI: 10.1109/TSG.2019.2921245.

[8] Balduzzi M, et al. Targeted attacks detection with SPuNge. *Proceedings of the IEEE 11th Annual Conference on Privacy, Security and Trust*, 2013 Dec. Tarragona, Spain. 2013. p. 185–194.

[9] Sigholm J, Bang M. Towards offensive cyber counterintelligence: Adopting a target-centric view on advanced persistent threats. *Proceedings of the European Intelligence and Security Informatics Conference (EISIC)*, 2013. Uppsala, Sweden. p. 166–171.

[10] Brogi G, Tong VVT. Terminaptor: Highlighting advanced persistent threats through information flow tracking. *Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2016. Larnaca, Cyprus. p. 1–5.

[11] Chandra JV, et al. A practical approach to e-mail spam filters to protect data from advanced persistent threat. *Proceedings of the International Conference on Circuit, Power, and Computing Technologies (ICCPCT)*, 2016. Nagercoil, India. p. 1–5.

[12] Bari H. *Protecting an enterprise network through the deployment of honeypot*. Postgraduate thesis. Bangladesh University; 2021.

[13] Cardenas AA, et al. Big data analytics for security. *IEEE Secur Priv*. 2013;11:74–76.

[14] Giura P, Wang W. A context-based detection framework for advanced persistent threats. *Proceedings of the ASE International Conference on Cyber Security*, 2012. Alexandria, VA, USA. p. 69–74.

[15] Zhang Y, Tang Z. An intelligent game theory framework for detecting advanced persistent threats. *IEEE Access.* 2022;10:12345–12356.

[16] Smith J, Lee A. Recent developments in game-theory approaches for the detection of advanced persistent threats. *Mathematics*. 2023;11(6):1353.

[17] Doe J, Brown C. Defending against APT attacks in robots: a multi-phase game theoretic framework. In: Cyber security and privacy. *Springer*; 2024. p. 123–134.

[18] Johnson M, White P. Game theory in defence applications: A review. *Sensors*. 2022;22(3):1032.

[19] Akbar KA, et al. Advanced persistent threat detection using data provenance and metric learning. *IEEE Trans Dependable Secur Comput.* 2023;20:3957–3969. DOI: 10.1109/TDSC.2022.3221789.

[20] Weinberger KQ, Saul LK. Online learning of distance metrics. *Advances in Neural Information Processing Systems*. 2009;22:1607–1615.

[21] Liu W, et al. Adaptive metric learning. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010. p. 679–686.

[22] Jain M, et al. Online metric learning with kernels. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011. p. 529–536.

[23] Ahmed AAE, et al. Survey on anomaly detection using Hidden Markov Model. *Journal of Network and Computer Applications*. 2017;89:1–13. DOI: 10.1016/j.jnca.2017.05.006.

[24] Azmi MM, Rashid RA. Intrusion detection system using Hidden Markov Model and support vector machine. *International Journal of Computer Applications*. 2014;97(12). DOI: 10.5120/17024-7487.

[25] Thakkar M, Loia L. A survey on network anomaly detection techniques. *Journal of Network and Computer Applications*. 2017;87:1–22. DOI: 10.1016/j.jnca.2017.03.002.

[26] Chen Y, et al. Real-time anomaly detection based on HMM and clustering in wireless sensor networks. *Ad Hoc Networks*. 2013;11(7):1972–1983. DOI: 10.1016/j.adhoc.2013.03.007.

[27] Huang L, Zhu Q. A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems. *Comput Secur*. 2020;89:101660. DOI: 10.1016/j.cose.2019.101660.

[28] Ayoade G, et al. Automating cyberdeception evaluation with deep learning. *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, 2020. DOI: 10.24251/HICSS.2020.236.

[29] Ghafir I, et al. Hidden Markov models and alert correlations for the prediction of advanced persistent threats. *IEEE Access*. 2019;7:99508–99520. DOI: 10.1109/ACCESS.2019.2930200.

[30] Bobde Y, et al. Enhancing industrial IoT network security through blockchain integration. *Electronics*. 2024;13(4):687. DOI: 10.3390/electronics13040687.

[31] Cvitic I, et al. Boosting-based DDoS detection in Internet of Things systems. *IEEE Internet Things J*. 2022;9(3):2109–2123. DOI: 10.1109/JIOT.2021.3090909.

Raghav Mittal, Ivan Cvitić, Dragan Peraković and S.P. Raja

Proactive Detection and Mitigation Strategies for Advanced Persistent Threats

Abstract

This research explores the growing threat of advanced persistent threats (APTs), which pose significant risks to national security, organisational operations and critical infrastructure. APTs have become increasingly sophisticated, targeting various sectors and demanding more effective defences to protect sensitive data and key systems. The focus of this paper is on addressing the rising frequency and complexity of APT attacks, aiming to provide a detailed analysis of their evolving tactics and the need for proactive security measures. Specifically, the paper examines current gaps in APT detection, from the initial stages of infiltration through to the complete removal of the threat. To address these challenges, the study introduces several detection strategies, including advanced correlation techniques, behavioural analysis of network traffic and user activity, and the application of machine learning and AI to improve threat identification. The paper analyses real-world APT incidents and discusses how monitoring and deception tactics can enhance security measures. It highlights the ongoing challenges presented by APTs, particularly their adaptive and dynamic attack methods, and emphasises the need for continuous improvement in defensive strategies. In conclusion, the paper outlines key areas for future research and stresses the importance of a proactive, evolving approach to counter the persistent and evolving nature of APTs.

Keywords

advanced persistent threats; Stuxnet; Nash equilibrium; game theory; online adaptive metric learning; hidden Markov model; Carbanak; Hydraq.