Advanced Persistent Threats (APT) Attribution Using Deep Reinforcement Learning

ANIMESH SINGH BASNET, Cyber Security Research Centre, London Metropolitan University, UK
MOHAMED CHAHINE GHANEM*, Cybersecurity Institute, Department of Computer Science, University of Liverpool & Cyber Security Research Centre, London Metropolitan University, UK
DIPO DUNSIN, Cyber Security Research Centre, London Metropolitan University, UK
HAMZA KHEDDAR, LSEA Laboratory, University of Medea, Algeria
WIKTOR SOWINSKI-MYDLARZ, Cyber Security Research Centre, London Metropolitan University, UK

Abstract: This paper investigates the application of Deep Reinforcement Learning (DRL) for attributing malware to specific Advanced Persistent Threat (APT) groups through detailed behavioural analysis. By analysing over 3,500 malware samples from 12 distinct APT groups, the study utilises sophisticated tools like Cuckoo Sandbox to extract behavioural data, providing a deep insight into the operational patterns of malware. The research demonstrates that the DRL model significantly outperforms traditional machine learning approaches such as SGD, SVC, KNN, MLP, and Decision Tree Classifiers, achieving an impressive test accuracy of 94.12%. It highlights the model's capability to adeptly manage complex, variable, and elusive malware attributes. Furthermore, the paper discusses the considerable computational resources and extensive data dependencies required for deploying these advanced AI models in cybersecurity frameworks. Future research is directed towards enhancing the efficiency of DRL models, expanding the diversity of the datasets, addressing ethical concerns, and leveraging Large Language Models (LLMs) to refine reward mechanisms and optimise the DRL framework. By showcasing the transformative potential of DRL in malware attribution, this research advocates for a responsible and balanced approach to AI integration, with the goal of advancing cybersecurity through more adaptable, accurate, and robust systems.

CCS Concepts: • **Security and privacy** → *Malware and its mitigation*; **Systems security**.

Additional Key Words and Phrases: **Key Words:** Advanced Persistent Threat (APT), Malware Attribution, AI, MDP, Deep Reinforcement Learning (DRL)

ACM Reference Format:
Animesh Singh Basnet, Mohamed Chahine Ghanem, Dipo Dunsin, Hamza Kheddar, and Wiktor Sowinski-Mydlarz. 2025. Advanced Persistent Threats (APT) Attribution Using Deep Reinforcement Learning. 1, 1 (May 2025), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

Authors' Contact Information: Animesh Singh Basnet, anb1380@my.londonmet.ac.uk, Cyber Security Research Centre, London Metropolitan University, London, UK; Mohamed Chahine Ghanem, mohamed.chahine.ghanem@liverpool.ac.uk, Cybersecurity Institute, Department of Computer Science, University of Liverpool & Cyber Security Research Centre, London Metropolitan University, London, UK; Dipo Dunsin, d.dunsin@londonmet.ac.uk, Cyber Security Research Centre, London Metropolitan University, London, UK; Hamza Kheddar, kheddar.hamza@univ-medea.dz, LSEA Laboratory, University of Medea, Medea, Algeria; Wiktor Sowinski-Mydlarz, w.sowinskimydlarz@londonmet.ac.uk, Cyber Security Research Centre, London Metropolitan University, London, UK.

Manuscript submitted to ACM

1 Introduction

In recent years, cyber-attacks have evolved from isolated incidents into sophisticated operations conducted by well-resourced Advanced Persistent Threats (APTs), which are characterised by their strategic, long-term approaches compared to more opportunistic cyberattacks [22]. Like traditional cyberattacks, APTs utilise malware as their primary tool, but they stand out due to their complexity, higher number of network events, and intricate behavioural activities [18]. APTs are meticulously orchestrated, employing advanced techniques to remain hidden while they extract data, disrupt operations, or create entry points for future attacks [49]. Often backed by nation-states or large organisations with political or economic motives, APTs pose significant threats to critical resources [22]. A notable example is the Stuxnet worm, which, although discovered in 2010, had been operating covertly since at least 2005, specifically targeting Iran's nuclear facilities at the Natanz uranium enrichment plant [4]. Developed by the USA, Stuxnet utilised advanced evasion techniques like zero-day exploits and rootkits to infiltrate and compromise its target while remaining undetected for years [4].

According to Statista, the global revenue from the APT protection market is expected to reach $12.5 billion by 2025, driven by the urgent need to defend against these evolving threats [23] [21]. Despite significant investments in security solutions, APT incidents, including ransomware attacks, continue to rise across industries, military sectors, and government institutions, with a 55.5% increase in ransomware cases in 2023 alone, reaching 4,368 incidents worldwide [33, 41]. The use of advanced technologies like large language models (LLMs) has further intensified the threat landscape, enabling more sophisticated cyberattacks [20]. This escalation underscores the critical need for innovative defence strategies, encouraging organisations and governments to continuously invest in advanced security measures to stay ahead of these persistent adversaries [5].

The increasing sophistication and frequency of APTs highlight the critical challenge of precise attribution in the cybersecurity landscape [18] [52] [21]. Accurate attribution is essential for developing targeted defensive strategies, as understanding an adversary's tactics, techniques, and procedures (TTPs) allows for tailored responses to specific threats [38] [52]. It also plays a key role in holding perpetrators accountable, which can act as a deterrent through legal and diplomatic consequences, thereby maintaining global cyber stability [30]. However, attribution is complicated by the obfuscation methods used by APTs, including routing attacks through proxies and deceptive indicators [43] [52]. These sophisticated tactics require extensive technical expertise and collaboration across sectors to analyse threat profiles that reveal attackers' motives and strategies [43]. Building on this complexity, each APT group possesses a distinct signature, merging specific malware applications with strategic objectives, whether for financial gain or disrupting critical infrastructure [23]. The intricacies of these profiles underscore the importance of attribution, pinpointing the perpetrator not only aids in defence but also in shaping cybersecurity policies and measures to pre-empt future attacks [9].

To address the growing challenge of attributing APTs, this report suggests leveraging machine learning algorithms that focus on analysing malware behaviour within sandbox environments [37]. Machine learning's ability to process vast datasets and detect subtle patterns offers a promising solution to understanding the complex and often hidden techniques used by APT groups [38]. By training models on behavioural data obtained from executing malware in virtual systems, these systems can be developed to automatically detect and classify patterns, leading to more precise attribution of cyberattacks. Within this context, Deep Reinforcement Learning (DRL) emerges as particularly effective for attributing APT malware [31]. DRL combines deep learning's pattern recognition capabilities with reinforcement learning's adaptive decision-making through trial and error, enabling it to detect and enhance its response to evolving

malware behaviours [32] [36]. Techniques like the Markov Decision Process (MDP) and model-free learning allow DRL to structure decision-making and adapt without relying on predefined models. Unlike traditional machine learning models that may struggle with the dynamic nature of cyber threats, DRL continuously learns and refines its strategies, making it highly effective against sophisticated APT tactics. Its ability to operate in environments with incomplete information, simulate diverse attack scenarios, and evolve through interaction underscores its potential as a powerful tool in crafting robust cyber defence strategies [34].

2 Related Work

The adoption of Deep Reinforcement Learning (DRL) for malware detection is a relatively recent and promising development in the wider field of cybersecurity [3]. Understanding the related work in this domain requires a foundational grasp of malware behaviours, origins, and classifications. Since malware plays a central role in APT attacks, analysing its behavioural patterns provides crucial insights into the tactics and identities of APT groups [55]. The works reviewed in this section explore the use of reinforcement learning in cybersecurity more broadly, followed by focused discussions on malware behavioural analysis, attribution, and family classification, critically assessing the effect of these elements to enhance the precision and efficiency of cybersecurity defences [15].

2.1 Reinforcement Learning in Cybersecurity

Reinforcement learning (RL) has long played a role in cybersecurity, with recent advances—particularly in deep reinforcement learning (DRL)—expanding its applications across a wide range of security tasks. Notably, DRL has been leveraged for network intrusion detection and anomaly detection, enabling adaptive identification of malicious activities in dynamic environments [39]. One approach integrates DRL with optimized feature selection to continuously recognize network attacks, resulting in improved detection rates on modern benchmark datasets [39]. These RL-driven intrusion detection systems can adjust to evolving attack patterns on the fly, generalizing beyond known signatures to identify previously unseen threats[11]. Unlike static, rule-based detectors, RL-based defenders learn policies through ongoing interaction with network traffic, allowing them to autonomously refine their detection strategies as new threats emerge [11].

Another active area of research in reinforcement learning for cybersecurity is its application to proactive defence mechanisms such as phishing prevention and honeypot deployment. In phishing detection, recent approaches utilize double deep Q-networks (DDQNs) to frame the task as a sequential decision-making problem, enabling systems to adapt dynamically to concept drift and class imbalances in malicious URLs [29]. This adaptability allows DRL-based classifiers to outperform traditional deep learning models, particularly on newer phishing datasets that reflect evolving attack strategies [29]. Similarly, RL has been applied to enhance honeypot systems through Deep Adaptive RL for Honeypots (DARLH) framework where agents learn optimal honeypot behaviours (in single-agent and multi-agent settings) to better engage and analyze attackers in real-time [11]. These systems autonomously adjust their responses in real-time based on attacker interactions, improving their ability to engage intruders and gather relevant threat intelligence as attacks unfold.

Together, these applications highlight the growing importance of reinforcement learning in building resilient, adaptive defences. By continuously learning from adversarial environments, RL-based systems can refine their strategies without manual intervention—making them especially effective for intrusion detection, phishing mitigation, and other security tasks where flexibility and real-time responsiveness are critical to countering sophisticated, fast-changing threats [45].

2.2 Behavioural Analysis

Building on the adaptive detection capabilities offered by RL, malware behavioural analysis serves as a cornerstone technique in cybersecurity that involves observing and understanding the actions performed by malware within a controlled environment, typically a sandbox [57]. This technique allows for the identification of malicious patterns and behaviours including networks and operations within the system. Recent studies emphasise the evolution of this analysis to include automated systems that leverage machine learning to predict and react to malware behaviour dynamically [17]. Such systems can discern between benign and malicious processes by examining changes made by the software to the system's state or its network behaviour [53]. These analyses often involve the extraction of features such as API calls, file-system operations, and network activity which are then processed using advanced algorithms to detect anomalous patterns that suggest malicious intent [53]. By comprehensively understanding the behaviour exhibited by the malware during execution and examining its underlying code and structure, we can gain valuable insights that aid in accurately attributing the malware to specific APT groups or threat actors [10].

2.3 Malware Attribution

The attribution of malware encompassing identifying the probable origin or actor behind an attack, is a complex yet crucial task within cybersecurity. Traditional approaches in malware attribution have relied heavily on manual, domain-specific feature engineering and pre-processing to isolate attributes indicative of a malware's lineage or family ties [40]. The incorporation of neural networks has significantly advanced malware attribution capabilities, particularly using machine learning techniques such as Random Forest and Extreme Gradient Boosting (XGBoost), which have strengthened efforts in this area [19]. These algorithms refine neural network models by improving accuracy and handling overfitting, essential for distinguishing between benign and malicious activities in vast and complex datasets [19]. This synergy optimises feature selection and enhances predictive capabilities, effectively supporting the identification of malware origins and behaviours.

Recent studies have introduced novel approaches in malware attribution that significantly improve upon traditional methods. For instance, the work by Binhui Tang and colleagues transforms APT malware samples into RGB images rather than relying on standard grayscale feature extraction [48] [36]. This approach allows for deeper and more nuanced feature mining, using an enhanced Convolutional Neural Network (CNN) model that incorporates Self-Attention mechanisms and Spatial Pyramid Pooling (SPP-net) [48]. This novel framework aids not only in detecting APT malware but also in facilitating the identification of malware origins and attack methodologies through sophisticated visual data representations. Another innovative approach is presented by Elijah Snow and his team, who utilised an end-to-end multimodal learning strategy [47]. This method integrates three distinct neural network architectures—dense networks, CNNs, and Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells—to automatically extract and learn features from diverse malware data attributes [47]. By combining these architectures, their model effectively classifies malware into respective groups, enhancing the granularity and accuracy of malware attribution. Further, Gil Shenderovitz and Nir Nissim introduced a dynamic analysis technique for segmenting Multivariate Time Series Data (MTSD) derived from API calls [46]. Their approach uses temporal segmentation to provide a detailed behavioural profile of APT malware, facilitating the detection and attribution to specific cyber-groups or nations with enhanced explainability [16].

Table 1. Summary of Recent Research Works on Malware Attribution

Research Work	Year	Dataset	Type of Data Used	RGB Image Conversion	Time Series Analysis	Feature Engineering	Multi-Input / Fusion Model	Machine Learning Model
Tang, et al. [48]	2023	3594 malware samples from 12 APT groups	Visual Data	X	X	X	X	Enhanced CNN with Self-Attention and SPP-net
Snow, et al. [47]	2020	Microsoft Malware Challenge Dataset	Multimodal Data	X	X	X	X	Dense Networks, CNN, RNN using LSTM
Shenderovitz, et al. [46]	2024	API calls of 12,655 malware from 188 APT groups	Time Series Data	X	✓	X	X	Machine Learning Classification
Zhang, et al. [56]	2024	Behavioural data and binary instructions from 2809 malware samples	Behavioural & Binary Data	✓	X	✓	✓	GNNs and ImageCNTM
Li, et al. [25]	2021	Behavioural data from 2389 malware samples from 7 APT groups	Behavioural Data	X	X	✓	X	Multiclass SMOTE-RF Model
Our System	2025	3594 malware samples from 12 APT groups	Behavioural Data	X	X	✓	X	Deep Learning Model

The work of Jian Zhang and colleagues improved the integration of multiple feature dimensions by employing a Graph Neural Network (GNN) model to create an event behaviour graph based on API instructions and operations, combined with an innovative ImageCNTM for capturing local spatial correlations and long-term dependencies of opcode images [56]. By fusing word frequency and behavioural features in a multi-input deep learning model, they propose a comprehensive system that classifies and accurately attributes APT malware, improving upon traditional single-dimensional models. Similarly, Shudong Li and his team refined the classification methodology by implementing a dynamic analysis and pre-processing stage for malware samples, followed by feature representation using the TF-IDF method and feature dimensionality reduction using the chi-square test [25]. Their innovative use of a Multiclass SMOTE-RF model addresses class imbalance, enhancing the classification accuracy significantly across various malware families [25]. Recent research by Mei and colleagues has proposed a deep learning-based network forensic framework for attributing APT attacks by analyzing encrypted network traffic and extracting key traceability features, which are then processed using a Multi-Layer Perceptual Deep Neural Network (MLP DNN) for anomaly detection [31]. When evaluated on the UNSW-NB15 dataset, this approach demonstrated superior performance in identifying and tracking APT events, surpassing traditional AI-based methods in both accuracy and forensic reliability [31].

However, despite these advancements, challenges persist, particularly when dealing with malware that lacks evolutionary links or belongs to completely different families. As noted by Rosenberg and colleagues, traditional methods often fall short in such scenarios because they primarily focus on detecting mutations or similarities within the same

functional group [40]. This limitation highlights the need for more advanced and flexible analytical tools to handle a broad spectrum of malware types, moving beyond familial or evolutionary similarities to embrace a more holistic and integrative approach in malware analysis [14].

Table 2. Comparative Analysis of Related Work on Leveraging Reinforcement Learning for ATP Detection

Research Work	Year	Data Source	Technique Used	Approach
Xuan & Cuong [8]	2024	Network traffic data involving APT IPs and normal IPs	(1) BiLSTM and Attention networks for unusual behaviour extraction in APT IPs. (2) Data rebalancing and contrastive learning for APT IP classification.	Introduces the FIERL model, combining advanced machine learning techniques to improve APT attack detection.
Saheed & Henna [41]	2023	Wireless network traffic data, including dynamic interactions and multi-stage APT attack patterns.	Deep Reinforcement Learning that dynamically interacts with the environment to learn and adapt to new APT attack strategies.	Proposed deep reinforcement learning to continually adapt and respond to evolving APT threats in wireless networks. This method outperforms traditional Feed Forward Neural Network models by learning faster
Atti & Yogi [3]	2024	Microsoft Malware Prediction Dataset	(1) Implements various techniques to prepare features extracted from executable files for training. (2) Involves data cleaning and preprocessing to optimise the dataset for model training. (3) Employs deep learning models, specifically Proximal Policy Optimization (PPO), to train the system's ability to detect malware.	Introduces a DRL framework for malware detection that learns complex patterns from executable files to identify malicious software

Addressing these challenges, this work presents a novel approach utilising Deep Reinforcement Learning (DRL) for malware attribution, specifically tailored for APTs developed by nation-states. DRL has previously shown significant advancements in malware detection, effectively identifying and responding to APT activities [31]. For instance, Cho Do Xuan and Nguyen Hoa Cuong have developed the FIERL model, which employs BiLSTM and Attention networks to extract unusual behaviour from network traffic data involving APT and normal IPs, further enhancing detection capabilities through data rebalancing and contrastive learning for APT IP classification [8]. Similarly, Kazeem Saheed and Shagufta Henna applied DRL to wireless network traffic data, where the system dynamically learns and adapts to new APT attack strategies, showcasing an ability to outperform traditional models by rapidly adjusting to evolving threats [41]. Additionally, Mangadevi Atti and Manas Kumar Yogi utilised a DRL framework that leverages Proximal Policy Optimization (PPO) to learn complex patterns from executable files, optimising malware detection processes and enhancing the system's predictive accuracy [3]. These applications underscore DRL's pivotal role in the detection domain, demonstrating its potential not only for identifying malware but also for attributing it effectively to specific APTs developed by nation-states.

DRL's application in the context of malware attribution offers a significant advancement over previous methods, as it does not rely on pre-defined models or static features, which are often limited by the need for extensive manual extraction and are less effective across disparate malware families [24, 54]. DRL leverages the strengths of deep learning for pattern recognition within complex and large-scale datasets, combined with the strategic decision-making capabilities of reinforcement learning [2]. This approach is particularly adept at processing incomplete or obfuscated data commonly employed in sophisticated cyberattacks, enabling it to adaptively learn and predict attribution based on behavioural patterns rather than static signatures [37].

3 Research Questions and Contribution

As previous sections have detailed the complexity and threat posed by APTs, this study leverages the sophisticated capabilities of DRL to analyse and interpret intricate malware data from controlled tests. The main goal is to refine a DRL model that effectively attributes APTs by analysing behavioural patterns, thus advancing cybersecurity defence mechanisms. This initiative to apply DRL seeks to harness its superior pattern recognition and strategic decision-making properties to enhance the detection and mitigation of advanced cyber threats.

3.1 Research Questions

The guiding questions of this research aim to critically evaluate the effectiveness of DRL in the cybersecurity landscape, particularly in attributing APTs. These questions explore: the identification of unique behavioural patterns of APTs within sandbox-analysed malware, the capability of DRL to precisely differentiate between malware behaviours from diverse APT groups, and the influence of the Markov Decision Process in boosting the strategic decision-making of DRL models within the context of cyber threat attribution. These inquiries are designed to assess whether DRL can offer a sophisticated and adaptive approach to understanding and countering APTs.

3.2 Contribution

This study makes impactful contributions to the domain of cybersecurity by pioneering the use of Deep Reinforcement Learning (DRL) for the specific purpose of APT attribution, benchmarking its effectiveness against traditional machine learning models, and exploring its adaptability to varied APT scenarios. It constructs a DRL model that not only processes and understands detailed behavioural data from malware but also empirically demonstrates its enhanced effectiveness over existing techniques. Additionally, by probing the model's ability to adapt to new threats, the research highlights DRL's potential to evolve and maintain relevance in a rapidly changing threat environment. The findings and methodologies of this research expand the practical and theoretical frameworks for deploying advanced AI in active cybersecurity defences, potentially setting new standards for the integration of machine learning in threat intelligence and response strategies.

4 Methodology

This section outlines the methodology adopted during the design, implementation and testing of the system. It provides details and justification on tools, approaches and methods employed as well as providing background information necessary for understanding the methodology.

4.1 Proposed System Design

The research adopts an experimental and simulation-based design, focusing on developing and evaluating a Deep Reinforcement Learning model for malware attribution to APT groups. The study begins by preparing a dataset of malware samples, followed by data pre-processing and feature extraction to ensure accuracy and relevance. The DRL model is then trained and tested in a simulated environment designed to mimic real-world conditions. This allows for controlled experimentation, where the model's ability to handle complex and evasive malware behaviours can be systematically assessed using metrics such as accuracy, robustness, and computational efficiency.

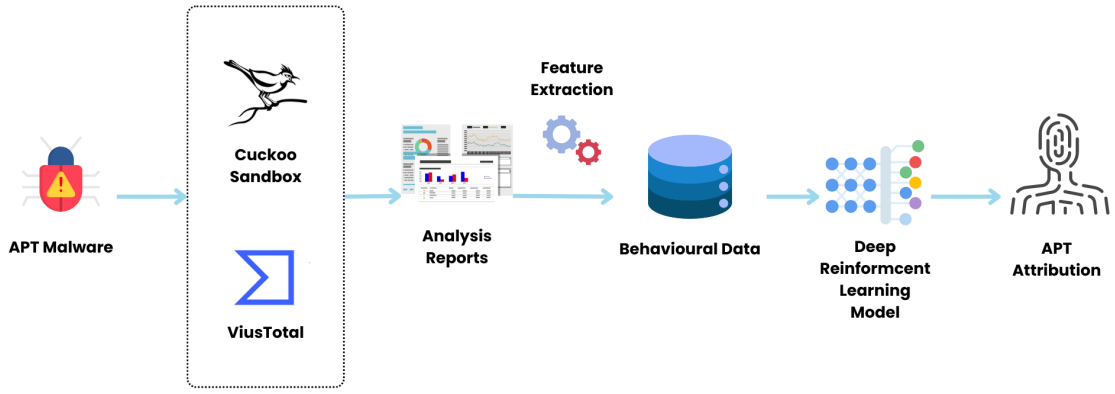

Fig. 1. System workflow for the proposed APT Attribution Model

4.2 Dataset

The APT Malware Dataset utilised in this work is a comprehensive collection of over 3,500 malware samples (https://github.com/cyber-research/APTMalware), categorised into 12 distinct Advanced Persistent Threat (APT) groups obtained from. These groups are believed to be state-sponsored by five different countries, including China, Russia, North Korea, the USA, and Pakistan. The dataset serves as a critical resource for benchmarking various machine-learning techniques aimed at authorship attribution of cyberattacks.[7]

Each APT group in the APT Malware Dataset represents a unique threat actor with a specific set of malware samples attributed to their cyber activities. The dataset's diversity is evident in both the number of samples per group and the variety of file types, ranging from executable files like .dll and .exe to documents such as .doc, .xlsx, and .ppt. This assortment adds complexity to the analysis, enabling robust evaluations of the various attack vectors and infection methods used by these groups. However, the dataset also shows a significant class imbalance, with some groups having as few as 32 samples and others as many as 961, presenting a challenge for reinforcement learning models to achieve unbiased behavioural representations. To manage this, the samples are meticulously labelled with their SHA-256 hashes for precise identification and stored in separate, password-protected compressed folders to ensure security and data integrity, with the universal password "infected" providing controlled access.

Table 3. APT Malware Dataset Distribution [7]

Country	APT Group	Sample Size
China	APT 1	405
China	APT 10	244
China	APT 19	32
China	APT 21	106
Russia	APT 28	214
Russia	APT 29	281
China	APT 30	164
North-Korea	DarkHotel	273
Russia	Energetic Bear	132
USA	Equation Group	395
Pakistan	Gorgon Group	961
China	Winnti	387
Total		**3594**

4.3 Data Collection

Data collection is strategically executed from two specialised sources to capture a broad spectrum of malware behaviours and characteristics, ensuring the depth and breadth of data necessary for effective DRL modelling.

4.3.1 Cuckoo Report. The Cuckoo Sandbox is an advanced open-source malware analysis system designed to analyse and report on the behaviour of potentially malicious files in a secure, isolated environment [6, 50]. It is widely used for malware detection by providing a controlled setting where files can be executed to observe their actions without risking the integrity of the host system [50]. In this work, the Cuckoo Sandbox plays an integral role in the hybrid process of collecting malware behavioural data, combining both manual and automated tasks to efficiently analyse large datasets of malware samples.

Since this analysis uses the web version of Cuckoo Sandbox, malicious files are manually uploaded through its interface, which offers system details and analytics while securely monitoring the files to protect the host system. During analysis, process IDs are generated to enable tracking and retrieval of detailed reports on each sample's activity, including system changes and network traffic. An automated script then extracts these IDs, structures the dataset, and removes duplicates to maintain data integrity and accuracy.

4.3.2 VirusTotal Report. VirusTotal is a comprehensive online service that analyses files and URLs to detect viruses, worms, trojans, and other kinds of malicious content [42, 51]. Leveraged by security professionals and researchers, VirusTotal aggregates information from over 70 antivirus scanners and URL/domain blacklisting services, along with a plethora of tools for the analysis of files, which makes it an indispensable resource for the real-time detection of emerging threats [42].

In this work, VirusTotal complements Cuckoo Sandbox by providing antivirus scan results and behaviour reports to enhance malware analysis. Automated scripts use each sample's SHA-256 hash to query the VirusTotal API and retrieve detailed file and behaviour reports. The file report includes detections, file type, size, and detection names, while the behaviour report highlights actions like registry changes and network activity. These insights support threat assessment and help train deep reinforcement learning models to detect similar behaviours in future attacks.

4.4 Data Understanding

Understanding the data collected from sources like Cuckoo Sandbox and VirusTotal is essential before diving into deeper analyses or model development, as it establishes the groundwork for recognising patterns, anomalies, and intrinsic properties of malware behaviours. This preliminary step ensures that subsequent processes, such as data cleaning, preprocessing, and detailed exploratory analysis, are effectively tailored to the characteristics of the data. For instance, the "reports.json" file from Cuckoo Sandbox provides a wealth of information on malware activities through detailed logs of file creation, registry changes, and network connections. By parsing these entries, it is possible to discern the common tactics used by malware, such as communication strategies and system infiltration methods, which are crucial for identifying threat behaviours.

Similarly, VirusTotal's file and behaviour reports complement the data by providing insights into the malware's detectable characteristics and operational tactics within infected systems. These reports include critical metadata on the malware's type, the extent of its recognition across different security platforms (unique_sources), and its evasion techniques (packers). Additionally, behavioural data like registry modifications and network traffic from these reports help in understanding how malware interacts with and affects systems, highlighting potential persistence mechanisms or damage attempts. Through a comprehensive understanding of these datasets, it is possible to understand that the data used in modelling is accurate, reliable, and robust enough to develop effective machine-learning models that can attribute malware to specific APT groups, enhancing cybersecurity measures and threat intelligence.

4.5 Data Preparation

Data preparation is crucial for transforming raw data from Cuckoo Sandbox and VirusTotal into a structured format suitable for in-depth analysis and modelling. The process begins with data cleaning, which involves refining the datasets to highlight essential malware characteristics. For file reports, this includes isolating key attributes like "file_name", "apt_group", and "unique_sources", and quantifying the threat level by analysing entries classified as "malicious". Additionally, the "import_list" is parsed to assess the complexity of malware interactions. For behaviour reports, the focus is on dynamic interactions, such as the number of files written and registry keys manipulated, which provide insights into the malware's impact on system operations. Cuckoo reports are also processed to extract API call statistics from "api_stats", giving a detailed view of system interactions at the API level.

Table 4. Feature dataset obtained after extraction, ranked using PCA analysis

Feature
Syswow64 Module
Files Modified
Files Deleted as Administrator
...
...
Total Signatures
API DeleteURLCacheEntryA

Following data cleaning, the process moves to data integration, where the cleaned datasets are merged into a cohesive framework for unified analysis. The file and behaviour datasets are merged with the Cuckoo reports, with missing

entries filled with zeros to maintain numerical data integrity. This step is essential for creating a comprehensive dataset that aligns all aspects of the malware's behaviour, enabling more effective modelling and analysis.

4.6 Data Modelling

In the data modelling phase, several critical steps are undertaken to prepare the dataset for effective machine learning applications. The process begins with Variable Transformation, where the dataset variables are identified and categorised based on their data types. Numerical columns are separated from categorical columns to facilitate different preprocessing techniques suitable for each type. The "apt_group" column, serving as the target variable for the models, is meticulously handled to ensure it is excluded from the feature sets when present in numerical columns, preventing data leakage. Categorical variables are then transformed into integer codes using techniques like Label Encoding, converting nominal data into a format that is digestible for machine learning algorithms. This transformation is essential for preparing the data for accurate and efficient modelling, ensuring that all features are in a machine-readable form.

Following the transformation, the dataset undergoes data partitioning, class imbalance treatment, and normalisation to optimise it for model training and evaluation. SMOTE (Synthetic Minority Over-sampling Technique) is employed to address class imbalances within the dataset, synthesising new examples in the minority class to prevent model bias towards the majority class. To further refine the class boundaries and remove overlapping samples, Tomek Links are applied after SMOTE to eliminate borderline examples that contribute to class ambiguity. This combination improves the separation between classes and reduces noise, enhancing the model's ability to learn meaningful patterns. This ensures a balanced representation across classes, which is crucial for generalising the model effectively.

Table 5. APT Malware Dataset Split After SMOTE and Tomek Links Application

	Sample Size	Percentage
Train	8,061	70%
Test	3,455	30%
Total	**11,516**	**100%**

The data is then split into training and testing sets, using a 70-30 split, reserving 30% of the data for testing to ensure a comprehensive evaluation of model performance. A higher proportion for testing is particularly important in attribution tasks, where generalisation to unseen, diverse attack patterns is critical. Subsequently, normalisation is performed using MinMaxScaler, scaling all features to a uniform range to prevent any single variable from dominating due to its scale. This step is vital as it allows the machine-learning model to converge more rapidly during training. The normalised data is then carefully reformatted back into DataFrames, retaining the original column names for better traceability and clarity during model training and evaluation phases.

4.7 Model Building

In the model-building phase, a bespoke environment is crafted using the Gymnasium framework, tailored to the complexities of Advanced Persistent Threat (APT) data. This setup precisely defines the observation space based on the feature set derived from malware samples and the action space aligned with unique labels constructed using the number of APT groups. The environment facilitates the simulation of interaction sequences, rewarding the model for accurate predictions and resetting for new episodes as data points are iteratively processed.

During the training of the model, a Deep Q-Network (DQN) is utilised, and configured with adjustable learning rates and buffer sizes to optimise the learning curve. The model's performance is periodically assessed using key metrics such as accuracy and the F1 score, which aid in fine-tuning the training regimen. This dynamic approach ensures a balance between the exploration of new strategies and the exploitation of known effective tactics, enhancing the model's ability to make progressively more accurate malware classifications. Subsequent development includes hyperparameter tuning—adjusting the discount factor, exploration rate (epsilon), and mini-batch sizes—to enhance the learning process's efficiency and effectiveness. Training episodes are varied in length to reflect the complex nature of real-world APT scenarios better, preventing overfitting and improving generalisation. Moreover, regularization techniques like dropout and batch normalisation are integrated within the neural network architecture to mitigate the risk of overfitting by moderating less predictive features' influence and stabilising learning across different batches. Detailed performance analysis and error metrics are continuously collected and reviewed to identify the model's strengths and weaknesses, providing a clear direction for its ability to capture the APT groups.

4.8 MDP Model

The Markov Decision Process (MDP) provides a structured framework for understanding how an agent makes decisions while interacting with its environment [27]. In this work, the MDP framework is utilised to design and develop the DRL model for attributing malware to APT groups. The primary data sources for the model come from detailed reports generated by Cuckoo Sandbox and VirusTotal, which offer comprehensive behavioural analyses of malware samples. These reports provide a multi-dimensional view of each malware's characteristics and behaviour, which are crucial for defining the states, actions, and rewards in the MDP framework as listed below:

4.8.1 States Space. The state represents the current understanding of a malware sample based on its observed behaviours and characteristics. Each state is derived from a feature dataset that encapsulates various aspects of malware behaviour, such as file operations, registry changes, network activities, and other dynamic interactions recorded during the malware's execution. This dataset is constructed using key data points extracted from the Cuckoo and VirusTotal report. These features collectively form a comprehensive behavioural profile of the malware, encapsulating its operational tactics and techniques, which are used to define the current state in the MDP. This state representation serves as the foundation for the reinforcement learning model's decision-making process, enabling accurate attribution and classification of malware to specific APT groups.

4.8.2 Actions Space. In the MDP model, an action refers to the transition from analysing one malware sample to another within the dataset. Each action involves selecting a new malware sample from the dataset and performing the analysis to obtain its behavioural profile, thus transitioning the state of the MDP from the current malware profile to the next. This action reflects the decision-making process in identifying and comparing malware attributes across different samples, which is central to attributing them to specific APT groups.

4.8.3 Rewards. The reward in our proposed MDP model is implemented using a hybrid reward strategy that combines both extrinsic and intrinsic rewards to guide the agent. The extrinsic reward reflects attribution accuracy: when the agent correctly attributes a malware sample to its corresponding APT group based on observed behaviours, it receives a reward of $(+1)$; incorrect attributions yield (0). The magnitude of the reward is scaled based on the confidence level of the attribution and the criticality of correctly identifying specific APT-related malware, reflecting the importance of precision in cybersecurity measures. Complementing this, the intrinsic reward promotes exploration by granting

an additional reward of (+0.5) when the agent encounters a novel state—defined by unique behavioural features not previously seen. The total reward at each step is the sum of both components, encouraging accurate classification while fostering diverse state exploration to improve generalization across unseen data.

In summary, the dataset's behavioural features define the states of malware samples within the MDP framework. The DRL agent interacts with these states to attribute samples to APT groups, receiving a reward of +1 for correct classifications and 0 otherwise. To promote exploration, it also receives +0.5 when encountering a novel state with previously unseen behavioural features.

5 Implementation and Testing

5.1 Simulation Environment

The proposed DRL model for Advanced Persistent Threat (APT) attribution utilises a structured approach incorporating an environment for sequential decision-making, a Q-network for estimating the quality of actions, and a replay memory for learning from past experiences. Here, the agent's states are derived from comprehensive behavioural data extracted from malware reports, while actions represent decisions to attribute malware to specific APT groups. An action represented by $a_t = n$ indicates the model's prediction, where n corresponds to the malware associated with an APT group. The agent operates within this environment, aiming to optimise the cumulative rewards over time, where the rewards are aligned with the accuracy of the attribution to the correct APT group. This structure is designed to refine the agent's decision-making process and improve its policy through continuous learning and adaptation based on detailed malware behaviour analysis.

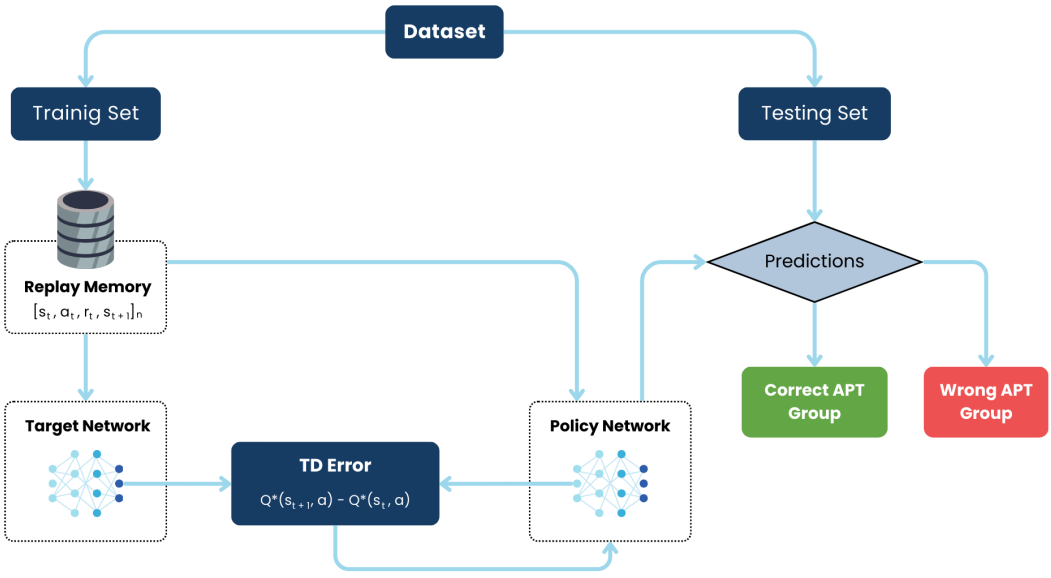

Fig. 2. DRL Model design for APT Attribution

DRL - Agent Policy Training Algorithm

Preconditions: $0 \leq \gamma \leq 1; 0.1 \leq \epsilon \leq 1$ 1: **Set** $X_\rho \leftarrow s_t$
2: **Set** $X_t \leftarrow s_{t+1}$
3: **For** each episode, **repeat**:
4: **While** $t \leq T$ **do**:
5: **If** $\epsilon \geq 0.1$ **then**:
6: **Select** random a_t /* ϵ-greedy strategy */
7: **Else if** $\epsilon \equiv 0.1$ **then**:
8: $Q(s_t, a_t) \leftarrow X_\rho \omega_\rho + b$
9: **Select** $a_t : a_t \leftarrow \text{index}(\max Q(s_t, a_t))$
10: **End If**
11: **Observe** r_t, s_{t+1}
12: **Store** experiences: $B_n \leftarrow \{(s_t, a_t, r_t, s_{t+1})_n\}$
13: **Select** randomly $B \subset B_n$
14: $Q(s_t, a_t) \leftarrow X_\rho \omega_\rho + b$
15: $Q(s_{t+1}, a_{t+1}) \leftarrow X_t \omega_t + b$
16: $Q^*(s, a) \leftarrow r_t + \gamma \max(Q(s_{t+1}, a_{t+1}))$
17: $L(fx) \leftarrow Q^*(s, a) - Q(s, a)$
18: $\omega_\rho \leftarrow \{\omega_\rho - \alpha \frac{dL}{d\omega_\rho}\}$ /* update weights */
19: **If** $T \equiv f$ **then**:
20: $X_t \omega_t + b \leftarrow X_p \omega_p + b$
21: **End If**
22: $t \leftarrow t + 1$
23: **End While**
24: **End For**

5.1.1 Environment. This is the simulated setting where the DRL agent is deployed, designed for making informed attributions of malware to specific Advanced Persistent Threat (APT) groups based on behavioural analysis. This environment is an adaptation of the OpenAI Gym interface, featuring a discrete action space that corresponds to different APT groups identified in the dataset [26]. The observation space is constructed from detailed features such as API calls, file-system operations, and network activities, which are crucial for defining the states of the malware being analysed [26].

5.1.2 Q-Network. At the core of the decision-making process, the Q-Network includes a policy network and a target network, each configured as a multilayer perceptron with two hidden layers leading to an output layer that represents each potential APT group. The networks use Leaky ReLU activation functions to maintain gradient flow during training, helping to prevent the vanishing gradient problem that can occur with standard ReLU functions if negative values are present in the inputs [35]. The output layers of the networks apply a MinMaxScaler to normalise the outputs, ensuring that the classification probabilities for the APT groups are scaled between 0 and 1 [44]. This normalisation helps stabilize the learning process by keeping the network's predictions within a consistent range.

5.1.3 Replay Memory. Essential for robust learning, Replay Memory archives tuples of the agent's experiences, including states, actions, rewards, and subsequent states. These experiences are accumulated as the agent processes

the behavioural data, employing an epsilon-greedy strategy ϵ to balance the exploration of new strategies with the exploitation of known patterns. Each action—representing an attribution decision—transitions the agent from one state to another (s_t to s_{t+1}), with rewards assigned based on the accuracy of these attributions.

5.1.4 Policy Training. The training of the DRL agent's policy operates over a series of episodes, with each episode consisting of numerous time steps, labelled as T. Each time step t involves the sampling of a feature vector representing the current state s_t from the replay buffer $\mathcal{B}$, which is then fed into the policy network. The policy network processes this input to output Q-values, $Q(s_t, a_t)$, for potential actions aimed at matching these values with the target or optimal Q-value, $Q(s, a)$.

Once the training process, detailed above, is complete, the efficacy of the agent's policy is evaluated by deploying the policy network model in a test environment. This test environment is carefully constructed using the validation dataset, allowing for a thorough assessment of the model's ability to perform under conditions that simulate real-world scenarios.

5.2 Experimental Specifications

The experimental setup for the DRL-based APT detection model involved careful tuning of the Deep Q-Network (DQN) parameters to optimize its ability to learn and adapt to advanced persistent threats (APTs). To train the DQN effectively for APT attribution, we implemented a dense reward system. This choice was driven by the discrete nature of the action space, where each action represents a classification decision. Dense rewards offered immediate feedback, enabling faster convergence and better credit assignment than sparse alternatives.

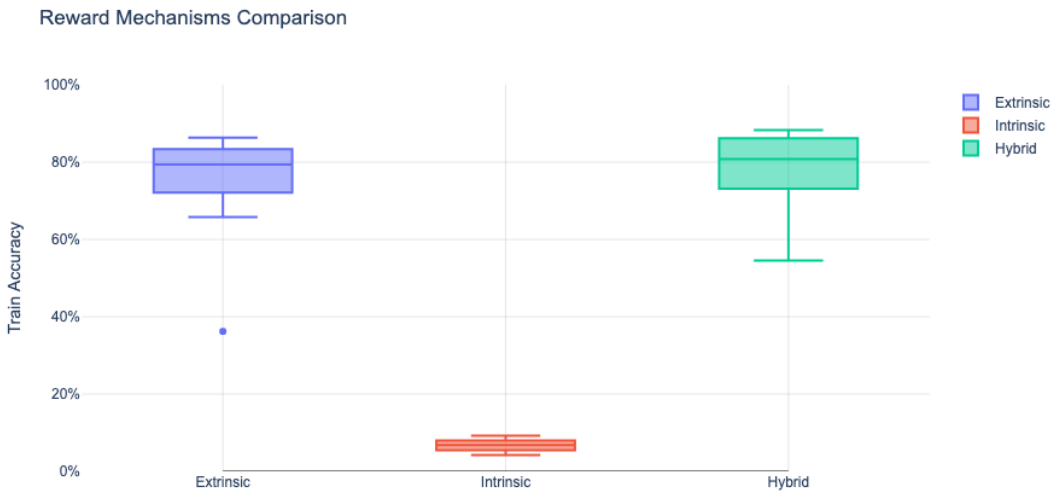

Fig. 3. Training Accuracy Comparison of Reward Mechanisms in the DQN Model

Next, we focused on reward design, which is a critical factor in reinforcement learning as it directly shapes agent behaviour. We experimented with extrinsic, intrinsic, and hybrid reward mechanisms. After evaluating their performance,

we chose the hybrid approach for its balanced support of accurate classification and exploratory behaviour. Although the differences narrowed later in training between hybrid and extrinsic, the hybrid method showed clear advantages early on, guiding its selection.

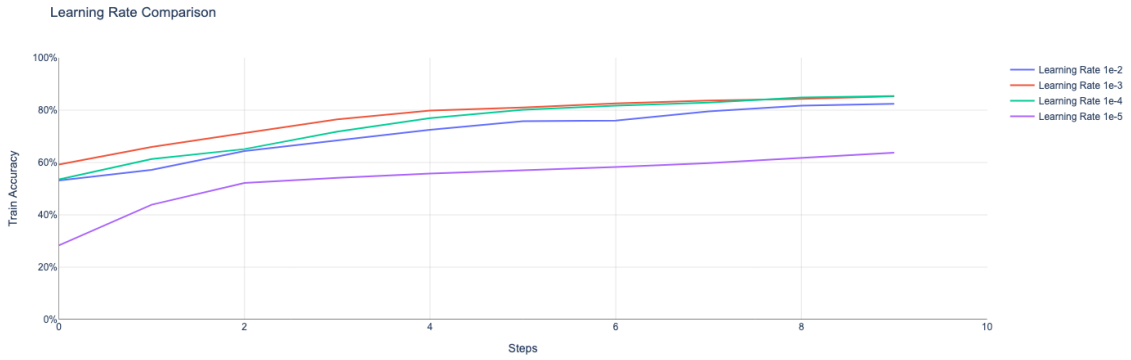

Fig. 4. Evaluating Learning Rates in the DQN Model Through Training Accuracy

For optimization, we tested a range of learning rates: 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}. Among these, 10^{-3} emerged as the most effective, offering stable training and high accuracy. The learning rate of 10^{-3} provided a sweet spot. It was low enough to keep training stable, yet high enough to allow rapid learning progress. The DQN's accuracy improved quickly in early training and smoothly converged to a high value without any volatile swings.

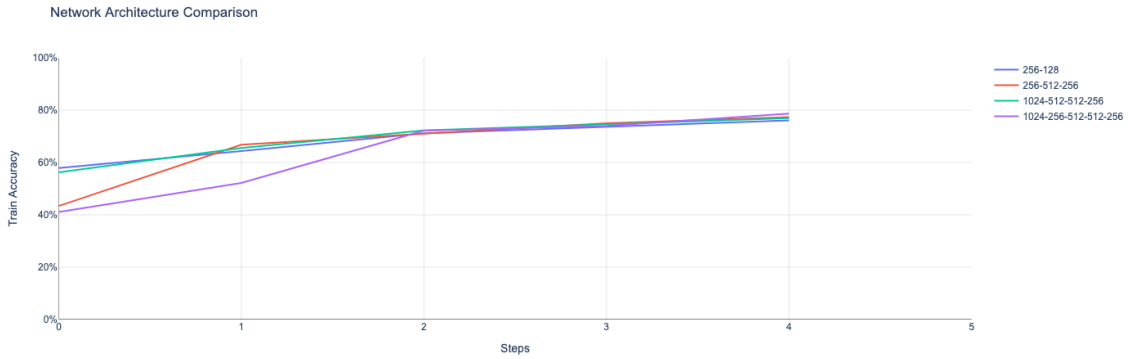

Fig. 5. Comparison of Network Architectures in the DQN Model Through Training Accuracy

Finally, we evaluated several policy network architectures, including [256, 128], [256, 512, 256], [1024, 512, 512, 256], and [1024, 256, 512, 512, 256]. The [1024, 512, 512, 256] architecture outperformed the others, providing a strong balance between capacity and generalization. While each architecture reached convergence over time, this model consistently showed better performance during the early training stages, which played a key role in its selection. These experiments informed the final configuration of the DRL-based APT detection model, summarized in the following comprehensive hyperparameter table.

Table 6. Experimental Setup for the DQN model

Parameter	Value
Learning Rate Schedule	$1 \times 10^{-3} \times (0.99 \text{ step}/1000)$
Policy	MlpPolicy
Buffer Size	100,000
Batch Size	256
Gradient Steps	3
Tau (τ)	0.005
Exploration Fraction	0.1
Exploration Final Eps	0.02
Gamma (γ)	0.99
Net Architecture	[1024, 512, 512, 256]
Activation Function	`torch.nn.LeakyReLU`

5.3 Software Environment

In order to implement and evaluate the DRL-based APT attribution model, several key software tools and techniques are utilised. These are essential for creating a robust environment that can simulate real-world scenarios and evaluate the performance of the model under controlled conditions.

5.3.1 Cuckoo Sandbox. Cuckoo Sandbox is an open-source automated malware analysis system that acts as a vital tool in the environment. It allows for the isolation and analysis of suspicious files in a safe, contained environment. This sandboxing technique enables the collection of detailed analysis about the behaviour of the file while running in an operating system, which is vital for training the DRL model to recognise threat behaviours. The outputs provided by Cuckoo Sandbox include API calls, network traffic, file system changes, and memory dumps, which serve as critical inputs for the model's learning process. [13]

5.3.2 Stable Baselines 3. Stable Baselines 3, an enhancement over the original OpenAI Baselines, offers refined implementations of reinforcement learning algorithms. The Deep Q-Network (DQN) model from Stable Baselines 3 is specifically utilised for the agent's training process. This model efficiently estimates the optimal action-value function, which is central to making informed decisions in the simulated network environments. While several DRL models were considered, including lightweight alternatives such as Dueling DQN and A2C, we conducted a comparative analysis using identical parameters and found that the standard DQN consistently outperformed the others. This superior performance is attributed to DQN's stability and efficiency in learning optimal action-value estimates, enabling the development of a robust policy capable of accurately distinguishing between benign and malicious network traffic in simulated environments. [1]

5.3.3 Gymnasium. Gymnasium, formerly known as Gym, is a tool from OpenAI that provides standardised interfaces for a diverse array of environments. These environments serve as testbeds for reinforcement learning algorithms. In this work, Gymnasium offers the foundational framework necessary for designing and managing the interaction between the DRL agent and the simulated network environment, which is vital for both the training and evaluation phases. It enables the DRL model to adapt and learn efficiently from dynamic scenarios that mimic real APT attacks. [12]

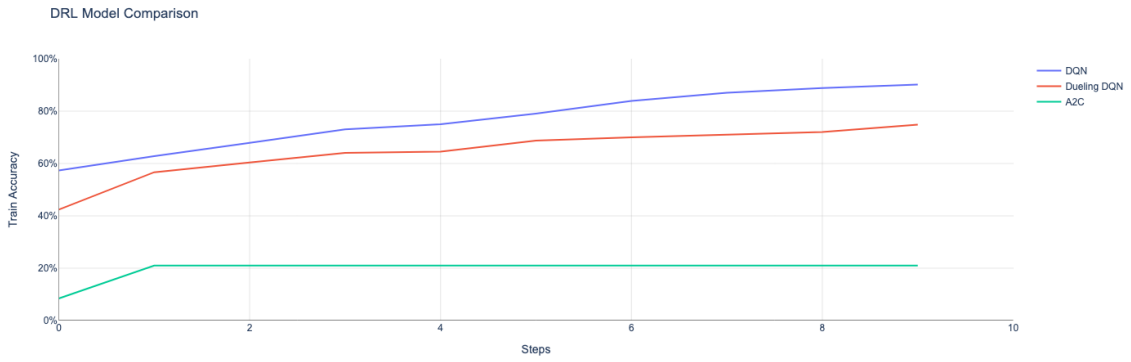

Fig. 6. Comparison of DRL Models Through Training Accuracy

6 Results and Discussions

6.1 Results Evaluation

The development of the DRL model for malware attribution involved extensive research, iterative coding, and numerous adjustments based on the insights gathered from predecessor models and contemporary research papers. This preparatory work was essential to establish a robust foundation for the model, ensuring it could adapt and respond effectively to the dynamic nature of malware threats. Initially, the model struggled with low accuracy levels, but through persistent adjustments to its architecture and learning algorithms, accuracy improved dramatically—from about 29% to over 79% in early iterations. By the end of the training, the model consistently reached accuracy levels near 98%, demonstrating its strong capability to accurately recognise and attribute malware activities. This upward trajectory in training accuracy is graphically represented in the Figure 7, which vividly illustrates the model's maturation and increasing proficiency over time.

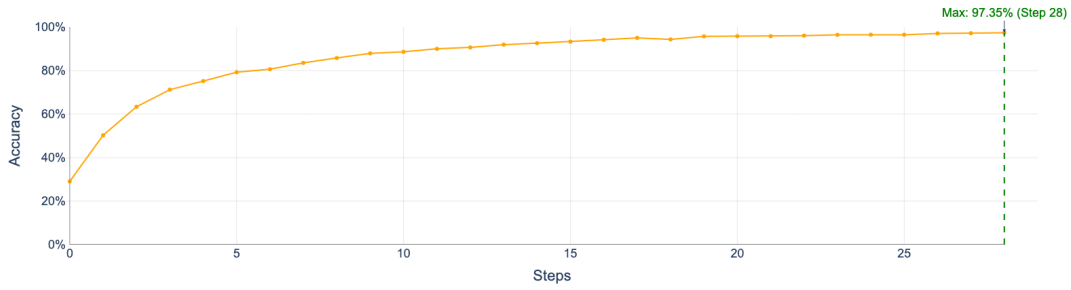

Fig. 7. Accuracy of the DRL Model for APT Attribution on Training Data

Following the graph for training accuracy, a detailed heatmap (Figure 8) was generated to gain insight into the model's performance across each of the APT groups, highlighting precision, recall, and F1 scores. Notably, the model demonstrated exceptional performance with 'Equation Group', achieving near perfect scores across all metrics, showcasing its

Fig. 8. Precision, Recall and F1 Score of the DRL Model for APT Attribution on each of the APT groups in the Training Data

capability to precisely attribute actions to this well-documented APT. Similarly, 'APT 19', 'Gorgon Group', and 'Energetic Bear' show remarkable precision and near-perfect F1 scores, reflecting the model's strength in handling sophisticated malware profiles. In contrast, 'APT 1' presents a lower recall of 90.27%, indicating a slight challenge in capturing all activities associated with this group. This variance in performance underscores areas for potential refinement, providing valuable feedback for further enhancing the model's accuracy and adaptability to diverse malware behaviours. This heatmap serves as a crucial tool for visualising the model's specific strengths and areas for improvement in malware attribution.

The overall performance of the DRL model on the test dataset can be quantified through key metrics including accuracy, precision, recall, and F1-score. The model achieved an accuracy of 94.12%, indicating a high rate of correctly identifying APT-related activities. Precision was obatined at 94.22%, suggesting that the majority of the model's predictions were relevant and accurately attributed to the correct APT groups. The recall rate of 92.07% reflects the model's ability to capture a substantial proportion of the actual positive cases, while the F1-score, 94.11%, illustrates a balanced relationship between precision and recall, confirming the model's robustness in various testing scenarios.

Table 7. Performance Metrics for the APT Attribution Model on the Test Dataset

Metric	Value
Accuracy	94.12%
Precision	94.22%
Recall	92.07%%
F1-score	94.11%%

Fig. 9. Precision, Recall and F1 Score of the DRL Model for APT Attribution on each of the APT groups in the Testing Data

The analysis of the DRL model's performance on the test dataset across various APT groups reveals its accuracy through precision, recall, and F1-scores as shown in Figure 9. The model excelled with 'APT 21' and 'Energetic Group', achieving F1 scores perfect scores, underscoring its proficiency in accurately identifying and attributing their activities. The model faced challenges with 'APT 1' and 'APT 28', where it recorded a lower recall of 85.03% and 89.17% respectively, indicating difficulties in correctly detecting this group's actions. This reduction in recall may be attributed to the higher similarity of their behavioural patterns to those of other APT groups, leading to occasional misclassifications. Additionally, potential class imbalance in the training data or greater variability in the tactics, techniques, and procedures (TTPs) used by these groups could have further contributed to the reduced detection performance.

In contrast, 'Energetic Bear' and 'Energetic Group' displayed almost perfect precision, highlighting the model's strength in pinpointing these groups with high accuracy. Generally, the model demonstrated relatively high precision, recall, and F1 scores across most groups, reflecting its overall effectiveness in accurately attributing a diverse array of APT activities.

6.2 Model Comparison

Following the evaluation of the DRL model's performance, it is essential to place its achievements in the context of alternative approaches. To establish a comprehensive understanding of the DRL model's capabilities, it was benchmarked against several other machine learning models that were developed using the same dataset. This comparative analysis is pivotal as it provides a clearer picture of the DRL model's relative efficiency and accuracy in attributing malware to specific APT groups. Models such as Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), and Decision Tree Classifier were implemented to represent a broad spectrum of machine learning techniques, each with its strengths and weaknesses in handling classification tasks. For consistency and fairness in comparison, all baseline models were configured using their default parameters as provided by the sci-kit-learn library, without any hyperparameter tuning.

Benchmarking was conducted using identical training and testing splits across all models to ensure that performance differences could be attributed solely to the model architecture and not to data variability. Standard evaluation metrics such as accuracy, precision, recall, and F1-score were employed to assess each model's performance comprehensively, allowing for an objective comparison against the DRL model.

Table 8. Comparison of test accuracy across different models, including the proposed DRL model

Model	Test Accuracy
SGD	72.50%
SVC	81.21%
KNN	88.05%
MLP	89.49%
Decision Tree Classifier	90.56%
DRL Model*	**94.12%**

The comparative analysis revealed that the DRL model significantly outperformed the other models, achieving a test accuracy of 94.12%. In contrast, the Decision Tree Classifier, which had the next highest accuracy, reached only 90.56%. Models such as MLP and KNN also showed strong performance with accuracies of 89.49% and 88.05% respectively, while the SVC and SGD trailed with 81.21% and 72.50%. The superior performance of the DRL model underscores its advanced capability in learning from and adapting to the complex patterns of malware behaviours more effectively than traditional models. This indicates not only the robustness of the DRL approach in handling the nuances of cybersecurity threat detection but also its potential to provide more reliable and precise attributions in real-world applications.

6.3 Limitations and Future Works

The study identifies several limitations in implementing the DRL model for APT attribution. One significant constraint is the high computational demand, as the DRL model requires extensive processing power and memory to handle

large datasets and perform complex computations. This resource-intensive nature can limit its scalability, particularly in environments with limited hardware capabilities. Additionally, the model's effectiveness heavily depends on the availability of high-quality and diverse training data. In cybersecurity, where data is often scarce or sensitive, this dependency can restrict the model's learning potential and adaptability. The complexity of implementing and fine-tuning the DRL model also poses a challenge; its sophisticated nature requires expert knowledge in both reinforcement learning and cybersecurity, along with careful parameter adjustments to maintain optimal performance, which can be both resource-intensive and a barrier to widespread adoption.

To address these limitations, future work could focus on enhancing the computational efficiency of the DRL model by refining its architecture, employing more efficient algorithms, and utilising techniques like transfer learning and model pruning to reduce computational load without sacrificing performance. Expanding the diversity of training datasets to include a broader range of malware samples would also strengthen the model's ability to generalise and improve accuracy across different attack types. Additionally, addressing legal and ethical considerations, such as data privacy and bias, should be a priority, with guidelines developed for the ethical use of AI in cybersecurity. Finally, leveraging Large Language Models (LLMs) could further enhance DRL systems by optimising reward mechanisms and decision-making strategies. LLMs can help create more dynamic reward structures, improving the balance between exploration and exploitation and ultimately boosting the model's capacity to detect and respond to complex security threats [28].

7 Conclusion

This research demonstrates the significant advancements in the application of Deep Reinforcement Learning (DRL) for attributing Advanced Persistent Threat (APT) groups, using a detailed dataset of over 3,500 malware samples across 12 distinct APT groups. The DRL model showcased its capabilities by significantly outperforming traditional machine learning approaches such as Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), K-Nearest Neighbours (KNN), Multi-Layer Perceptron (MLP), and Decision Tree Classifiers. With a remarkable test accuracy of 94.12%, the DRL model stands out, not only for its high precision in malware attribution but also for its adaptability to the complex and evolving landscape of cyber threats. By applying DRL, organisations can enhance their threat intelligence capabilities, allowing for more nuanced understanding and preemptive actions against APTs. This study's findings underscore the potential of DRL in enhancing cybersecurity operations by providing rapid and accurate threat attribution, paving the way for further research on its applicability across more diverse datasets and optimising its computational efficiency for broader use in real-world scenarios.

Research Ethics: This study was deemed exempt from ethics approval as it did not involve human or animal subjects.

Code and Data: The code and datasets used and generated during this research are made publicly available at https://github.com/crypticsy/APTAttribution

References

[1] Stable Baselines 3. 2024. Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations & x2014; Stable Baselines3 2.4.0a9 documentation — stable-baselines3.readthedocs.io. https://stable-baselines3.readthedocs.io/en/master/. [Accessed 20-08-2024].

[2] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38. https://doi.org/10.1109/MSP.2017.2743240

[3] Mangadevi Atti and Manas Yogi. 2024. Application of Deep Reinforcement Learning (DRL) for Malware Detection. *International Journal of Information technology and Computer Engineering* 4 (04 2024), 23–35. https://doi.org/10.55529/ijitc.43.23.35

[4] Marie Baezner and Patrice Robin. 2017-10-18. *Stuxnet*. Report 4. Zurich. https://doi.org/10.3929/ethz-b-000200661

[5] Robin Buchta, George Gkoktsis, Felix Heine, and Carsten Kleiner. 2024. Advanced Persistent Threat Attack Detection Systems: A Review of Approaches, Challenges, and Trends. *Digital Threats* (Sept. 2024). https://doi.org/10.1145/3696014 Just Accepted.

[6] cuckoosandbox. 2024. Cuckoo Sandbox - Automated Malware Analysis — cuckoosandbox.org. https://www.cuckoo.ee/; https://github.com/cuckoosandbox/cuckoo. [Accessed 11-10-2024].

[7] CyberResearch. 2019. GitHub - cyber-research/APTMalware: APT Malware Dataset Containing over 3,500 State-Sponsored Malware Samples — github.com. https://github.com/cyber-research/APTMalware. [Accessed 05-10-2024].

[8] Cho Do Xuan and Nguyen Hoa Cuong. 2024. A novel approach for APT attack detection based on feature intelligent extraction and representation learning. *PLoS One* 19, 6 (June 2024), e0305618. https://doi.org/10.1371/journal.pone.0305618

[9] Dipo Dunsin, Mohamed Chahine Ghanem, Karim Ouazzane, and Vassil Vassilev. 2025. Reinforcement learning for an efficient and effective malware investigation during cyber incident response. *High-Confidence Computing* (2025), 100299. https://doi.org/10.1016/j.hcc.2025.100299

[10] Mohammed Ashfaaq M Farzaan, Mohamed Chahine Ghanem, Ayman El-Hajjar, and Deepthi N Ratnayake. 2024. AI-Enabled System for Efficient and Effective Cyber Incident Detection and Response in Cloud Environments. *arXiv preprint arXiv:2404.05602* (2024).

[11] Salvo Finistrella, Stefano Mariani, and Franco Zambonelli. 2025. Multi-Agent Reinforcement Learning for Cybersecurity: Classification and survey. *Intelligent Systems with Applications* 26 (2025), 200495. https://doi.org/10.1016/j.iswa.2025.200495

[12] Farama Foundation. 2023. Gymnasium Documentation — gymnasium.farama.org. https://gymnasium.farama.org/index.html. [Accessed 19-08-2024].

[13] Neil Fox. 2023. Cuckoo Sandbox Overview — varonis.com. https://www.varonis.com/blog/cuckoo-sandbox. [Accessed 17-08-2024].

[14] Mohamed Chahine Ghanem, Elhadj Benkhelifa, Dominik Wojtczak, Mohamed Amine Ferrag, Norbert Tihanyi, and Erivelton Geraldo Nepomuceno. 2025. Leveraging Reinforcement Learning for an Efficient Automation of Windows Registry Analysis During Cyber Incident. (2025).

[15] Mohamed C Ghanem, Thomas M Chen, Mohamed A Ferrag, and Mohyi E Kettouche. 2023. ESASCF: expertise extraction, generalization and reply framework for optimized automation of network security compliance. *IEEE Access* (2023). https://doi.org/10.1109/ACCESS.2023.3332834

[16] Mohamed Chahine Ghanem, Patrick Mulvihill, Karim Ouazzane, Ramzi Djemai, and Dipo Dunsin. 2023. D2WFP: a novel protocol for forensically identifying, extracting, and analysing deep and dark web browsing activities. *Journal of Cybersecurity and Privacy* 3, 4 (2023), 808–829.

[17] Nana Kwame Gyamfi, Nikolaj Goranin, Dainius Ceponis, and Habil Antanas Čenys. 2023. Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review. *Applied Sciences* 13, 21 (2023). https://doi.org/10.3390/app132111908

[18] Weijie Han, Jingfeng Xue, Yong Wang, Zhenyan Liu, and Zixiao Kong. 2019. MalInsight: A systematic profiling based malware detection framework. *Journal of Network and Computer Applications* 125 (2019), 236–250. https://doi.org/10.1016/j.jnca.2018.10.022

[19] Md Hasan, Muhammad Usama Islam, and Jasim Uddin. 2023. Advanced Persistent Threat Identification with Boosting and Explainable AI. *SN Computer Science* 4 (03 2023). https://doi.org/10.1007/s42979-023-01744-x

[20] Mohammed Hassanin and Nour Moustafa. 2024. A Comprehensive Overview of Large Language Models (LLMs) for Cyber Defences: Opportunities and Directions. https://doi.org/10.48550/ARXIV.2405.14487

[21] Nadim Ibrahim, Rajalakshmi N R, and Karam Hammadeh. 2024. Exploration of Defensive Strategies, Detection Mechanisms, and Response Tactics Against Advanced Persistent Threats APTs. *Nanotechnology Perceptions* 20 (05 2024), 439–455. https://doi.org/10.62441/nano-ntp.v20iS4.33

[22] kasperskyLab. 2020. The power of threat attribution: Challenges and benefits of cyberthreat attribution — kaspersky.com. https://media.kaspersky.com/en/business-security/enterprise/threat-attribution-engine-whitepaper.pdf. [Accessed 12-05-2024].

[23] Edward Kost. 2023. What is an Advanced Persistent Threat (APT)? | UpGuard — upguard.com. https://www.upguard.com/blog/what-is-an-advanced-persistent-threat. [Accessed 04-08-2024].

[24] Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion* 85 (2022), 1–22. https://doi.org/10.1016/j.inffus.2022.03.003

[25] Shudong Li, Qianqing Zhang, Xiaobo Wu, Weihong Han, and Zhihong Tian. 2021. Attribution Classification Method of APT Malware in IoT Using Machine Learning Techniques. *Security and Communication Networks* 2021 (09 2021), 1–12. https://doi.org/10.1155/2021/9396141

[26] Xiaoguang Li. 2023. Create custom OpenAI Gym environment for Deep Reinforcement Learning (drl4t-04). https://lixiaoguang.medium.com/create-custom-openai-gym-environment-for-deep-reinforcement-learning-drl-af2b2e3c830d. [Accessed 15-08-2024].

[27] Manuel Lopez-Martin, Belen Carro, and Antonio Sanchez-Esguevillas. 2020. Application of deep reinforcement learning to intrusion detection for supervised problems. *Expert Systems with Applications* 141 (2020), 112963. https://doi.org/10.1016/j.eswa.2019.112963

[28] Fatma Yasmine Loumachi, Mohamed Chahine Ghanem, and Mohamed Amine Ferrag. 2025. Advancing Cyber Incident Timeline Analysis Through Retrieval-Augmented Generation and Large Language Models. *Computers* 14, 2 (2025), 67. https://doi.org/10.3390/computers14020067

[29] Antonio Maci, Alessandro Santorsola, Antonio Coscia, and Andrea Iannacone. 2023. Unbalanced Web Phishing Classification through Deep Reinforcement Learning. *Computers* 12, 6 (2023). https://www.mdpi.com/2073-431X/12/6/118

[30] Joaquin Matamis. [n. d.]. Advancing Accountability in Cyberspace • Stimson Center — stimson.org. https://www.stimson.org/2024/advancing-accountability-in-cyberspace/. [Accessed 05-09-2024].

[31] Yangyang Mei, Weihong Han, Shudong Li, Kaihan Lin, Zhihong Tian, and Shumei Li. 2024. A Novel Network Forensic Framework for Advanced Persistent Threat Attack Attribution Through Deep Learning. *IEEE Transactions on Intelligent Transportation Systems* 25, 9 (2024), 12131–12140. https://doi.org/10.1109/TITS.2024.3360260

[32] Eduardo C. Garrido Merchán. 2023. Why deep reinforcement learning is going to be the next big deal in AI — eduardogarrido90. https://medium.com/@eduardogarrido90/why-deep-reinforcement-learning-is-going-to-be-the-next-big-deal-in-ai-2e796bdf47d2. [Accessed 10-08-2024].

[33] The Hacker News. 2024. 3 Ransomware Group Newcomers to Watch in 2024 — thehackernews.com. https://thehackernews.com/2024/01/3-ransomware-group-newcomers-to-watch.html. [Accessed 06-08-2024].

[34] Sang Ho Oh, Jeongyoon Kim, Jae Hoon Nah, and Jongyoul Park. 2024. Employing Deep Reinforcement Learning to Cyber-Attack Simulation for Enhancing Cybersecurity. *Electronics* 13, 3 (2024). https://doi.org/10.3390/electronics13030555

[35] Juan C Olamendy. 2023. Understanding ReLU, LeakyReLU, and PReLU: A Comprehensive Guide — juanc.olamendy. https://medium.com/@juanc.olamendy/understanding-relu-leakyrelu-and-prelu-a-comprehensive-guide-20f2775d3d64. [Accessed 18-08-2024].

[36] Aghila Rajagopal, Gyanendra Prasad Joshi, A. Ramachandran, R. T. Subhalakshmi, Manju Khari, Sudan Jha, K. Shankar, and Jinsang You. 2020. A Deep Learning Model Based on Multi-Objective Particle Swarm Optimization for Scene Classification in Unmanned Aerial Vehicles. *IEEE Access* 8 (2020), 135383–135393. https://doi.org/10.1109/ACCESS.2020.3011502

[37] Nanda Rani, Bikash Saha, and Sandeep Kumar Shukla. 2024. A Comprehensive Survey of Advanced Persistent Threat Attribution: Taxonomy, Methods, Challenges and Open Research Problems. arXiv:2409.11415 [cs.CR] https://arxiv.org/abs/2409.11415

[38] Muhammad Raza. 2023. What Are TTPs? Tactics, Techniques & Procedures Explained | Splunk — splunk.com. https://www.splunk.com/en_us/blog/learn/ttp-tactics-techniques-procedures.html. [Accessed 10-08-2024].

[39] Kezhou Ren, Yifan Zeng, Zhiqin Cao, and Yingchao Zhang. 2022. ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model. *Scientific Reports* 12 (09 2022). https://doi.org/10.1038/s41598-022-19366-3

[40] Ishai Rosenberg, Guillaume Sicard, and Eli (Omid) David. 2018. End-to-End Deep Neural Networks and Transfer Learning for Automatic Analysis of Nation-State Malware. *Entropy* 20, 5 (2018). https://doi.org/10.3390/e20050390

[41] Kazeem Saheed and Shagufta Henna. 2023. Deep Reinforcement Learning for Advanced Persistent Threat Detection in Wireless Networks. In *2023 31st Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*. 1–6. https://doi.org/10.1109/AICS60730.2023.10470498

[42] Samet. 2022. What is Virus Total? — sametyorulmaz777. https://medium.com/@sametyorulmaz777/what-is-virus-total-70c64b7c5e95. [Accessed 11-10-2024].

[43] Tinshu Sasi, Arash Habibi Lashkari, Rongxing Lu, Pulei Xiong, and Shahrear Iqbal. 2023. A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges. *Journal of Information and Intelligence* (2023). https://doi.org/10.1016/j.jiixd.2023.12.001

[44] scikitLearn. 2024. MinMaxScaler — scikit-learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. [Accessed 16-08-2024].

[45] Mohit Sewak, Sanjay K. Sahay, and Hemant Rathore. 2022. Deep Reinforcement Learning in the Advanced Cybersecurity Threat Detection and Protection. *Information Systems Frontiers* 25, 2 (Aug. 2022), 589–611. https://doi.org/10.1007/s10796-022-10333-x

[46] Gil Shenderovitz and Nir Nissim. 2024. Bon-APT: Detection, attribution, and explainability of APT malware using temporal segmentation of API calls. *Computers & Security* 142 (2024), 103862. https://doi.org/10.1016/j.cose.2024.103862

[47] Elijah Snow, Mahbubul Alam, Alexander Glandon, and Khan Iftekharuddin. 2020. End-to-end Multimodel Deep Learning for Malware Classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*. 1–7. https://doi.org/10.1109/IJCNN48605.2020.9207120

[48] Binhui Tang, Peiyun Leng, Xuxiang Shen, and Yuhang Wei. 2023. Deep Learning-Based APT Malware and Variants Detection with Attribution Analysis. *2023 IEEE 3rd International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, 996–1003. https://doi.org/10.1109/PRAI59366.2023.10331977

[49] BinHui Tang, JunFeng Wang, Zhongkun Yu, Bohan Chen, Wenhan Ge, Jian Yu, and TingTing Lu. 2022. Advanced Persistent Threat intelligent profiling technique: A survey. *Computers and Electrical Engineering* 103 (2022), 108261. https://doi.org/10.1016/j.compeleceng.2022.108261

[50] Manasa Vemuri. 2020. Malware Analysis using Cuckoo Sandbox — easypeazyo14. https://medium.com/@easypeazyo14/malware-analysis-using-cuckoo-sandbox-756616e6e85e. [Accessed 11-10-2024].

[51] VirusTotal. 2024. VirusTotal — virustotal.com. https://www.virustotal.com/gui/intelligence-overview. [Accessed 11-10-2024].

[52] Yuntao Wang, Han Liu, Zhendong Li, Zhou Su, and Jiliang Li. 2024. Combating Advanced Persistent Threats: Challenges and Solutions. *IEEE Network* 38, 6 (2024), 324–333. https://doi.org/10.1109/MNET.2024.3389734

[53] Chaoxian Wei, Qiang Li, Dong Guo, Xiangyu Meng, and Angel M. Del Rey. 2021. Toward Identifying APT Malware through API System Calls. *Sec. and Commun. Netw.* 2021 (Jan. 2021), 14 pages. https://doi.org/10.1155/2021/8077220

[54] Nan Xiao, Bo Lang, Ting Wang, and Yikai Chen. 2024. APT-MMF: An advanced persistent threat actor attribution method based on multimodal and multilevel feature fusion. arXiv:2402.12743 [cs.CR] https://arxiv.org/abs/2402.12743

[55] Shui Yu, Guofei Gu, Ahmed Barnawi, Song Guo, and Ivan Stojmenovic. 2015. Malware Propagation in Large-Scale Networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 1 (2015), 170–179. https://doi.org/10.1109/TKDE.2014.2320725

[56] Jian Zhang, Shengquan Liu, and Zhihua Liu. 2024. Attribution classification method of APT malware based on multi-feature fusion. *PLoS One* 19, 6 (June 2024), e0304066. https://doi.org/10.1371/journal.pone.0304066

[57] Mohamad Fadli Zolkipli and Aman Jantan. 2010. Malware Behavior Analysis: Learning and Understanding Current Malware Threats. In *2010 Second International Conference on Network Applications, Protocols and Services*. 218–221. https://doi.org/10.1109/NETAPPS.2010.46

Received XX XX XXXX; revised XX XX XXXX; accepted XX XX XXXX