Adversarial Malware and AI Security: A Survey

www.surveyx.cn

Abstract

This survey paper explores the intricate dynamics of adversarial malware and AI security in the evolving landscape of cybersecurity. It underscores the dual role of artificial intelligence (AI) in enhancing threat detection and being exploited by adversarial tactics. The paper highlights the vulnerabilities of machine learning models to adversarial attacks, emphasizing the need for robust detection systems. It examines techniques employed by adversarial malware to evade detection, including obfuscation and dynamic code generation, and the integration of static and dynamic analysis methods to enhance detection accuracy. The survey also delves into AI-based detection techniques, such as image-based and graph-based methods, and the development of novel AI architectures to counteract sophisticated threats. Additionally, it discusses the implications of adversarial attacks for AI security, advocating for innovative defense mechanisms and adaptive strategies to bolster AI resilience. The paper concludes by identifying future research directions, including the development of adaptive models, the integration of bio-inspired computation, and the expansion of dynamic analysis capabilities. By addressing these challenges, the survey aims to bridge the gap in research regarding malware evasion techniques, paving the way for future advancements in cybersecurity.

1 Introduction

1.1 Significance of Adversarial Malware and AI Security

Understanding adversarial malware and AI security is critical due to the rapidly evolving nature of cybersecurity threats and the sophisticated evasion techniques employed by malware authors. The continuous adaptation of malware to bypass detection systems presents significant challenges, particularly as model-based detectors are vulnerable to adversarial examples. This vulnerability is intensified by the increasing complexity of polymorphic and metamorphic malware, which traditional detection methods often fail to identify [1].

AI plays a dual role in cybersecurity, enhancing defense mechanisms while also being exploited by malicious actors [2]. The susceptibility of deep learning algorithms to adversarial inputs that mislead malware classification highlights the need for a thorough understanding of these vulnerabilities [3]. Machine learning classifiers are particularly prone to adversarial examples and concept drift, resulting in misclassification and diminished trust in their outputs, especially in critical applications like malware detection [4].

The inherent vulnerabilities of AI models in malware detection make them prime targets for evasion attacks, which can misclassify malware as benign [5]. The security and reliability of machine learning frameworks are paramount, given the severe consequences of successful adversarial attacks. Understanding adversarial malware classification is essential for improving antivirus products and addressing evolving cybersecurity threats [5].

In mobile ecosystems, particularly on Android platforms, the proliferation of malicious applications poses significant threats to user security and privacy [6]. Malware authors employ obfuscation techniques that complicate detection efforts, necessitating the development of more robust AI-

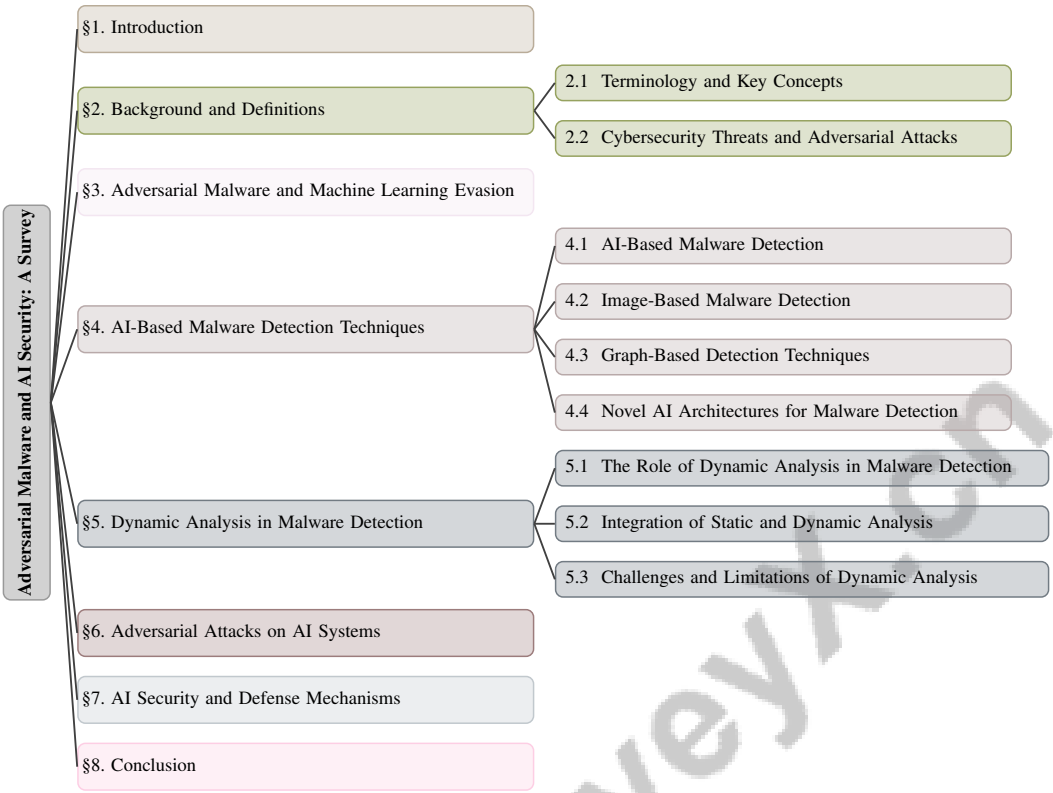


Figure 1: chapter structure

based security measures. As the cybersecurity landscape evolves, the importance of understanding adversarial malware and AI security becomes increasingly apparent, driving the need for innovative solutions to protect digital environments from these sophisticated threats [7].

1.2 Motivation for the Survey

This survey is motivated by the urgent need to tackle the challenges posed by sophisticated malware and the limitations of current detection methodologies. Traditional detection systems, often reliant on static signature-based approaches, have proven inadequate in identifying novel and complex malware variants, particularly zero-day threats. This inadequacy necessitates a shift towards advanced machine learning techniques that can enhance threat detection and defense capabilities [8].

The vulnerability of deep learning-based malware detectors to adversarial attacks emphasizes the need for robust detection models that can resist evasion tactics. The survey highlights innovative approaches like VisMal, which aim to address existing challenges in malware classification [5]. Furthermore, it emphasizes the necessity for adaptive defense strategies against evasion attacks, advocating for real-time responses to adversarial tactics [9].

Additionally, the survey addresses critical issues such as data leakage in mobile and IoT applications, focusing on Information Flow Control (IFC) techniques to detect and prevent unauthorized data access [10]. The complexities involved in analyzing script exploits and emergent behaviors during program execution further underscore the necessity of this comprehensive survey [11].

The survey aims to provide a principled approach to enhancing the robustness of malware detection systems against evasion attacks [12], particularly focusing on adversarial evasion techniques in the context of Windows PE malware files [7]. Moreover, capturing malware threat intelligence through an ontology allows for the structured representation of diverse Cyber Threat Intelligence (CTI) sources, facilitating better understanding and analysis [6].

Ultimately, this survey seeks to explore new approaches to enhance the safety and robustness of machine learning frameworks, as previous methods have struggled to adequately address vulnerabilities

[13]. By systematically analyzing these challenges, the survey aims to bridge the gap in research regarding the prevalence and evolution of malware evasion techniques, paving the way for future research and development in cybersecurity [14].

1.3 Relevance in the Current Cybersecurity Landscape

The survey's relevance is highlighted by the escalating complexity of cybersecurity threats, particularly the rise of polymorphic malware and advanced persistent threats (APTs), which challenge traditional security measures [15]. The continuous evolution of adversarial attacks underscores the inadequacies of conventional detection strategies, necessitating a comprehensive analysis of innovative solutions. Current methodologies, including static and dynamic approaches, face significant obstacles in identifying zero-day malware, especially within customized virtual machine environments [1]. This survey addresses these challenges by exploring advanced techniques that demonstrate high detection accuracy across diverse environments.

Integrating multi-view representations of malware, beyond traditional binary perspectives, is essential for enhancing detection capabilities, as current adversarial malware generation predominantly focuses on singular representations. The vulnerability of machine learning-based malware detectors to sophisticated evasion tactics, including adversarial attacks that manipulate input samples to evade detection, highlights the critical need for more resilient detection models capable of effectively combating the growing threat of malware cybercrime [16, 4, 17]. Additionally, dynamic analysis plays a crucial role in classifying malware into families, aiding in the identification of variants and enhancing the understanding of malware behaviors.

Recent trends in cybersecurity illustrate significant vulnerabilities in deep learning models, emphasizing the importance of this survey in comprehending adversarial malware. The increasing role of AI in cybersecurity and emerging threats from adversarial AI applications further underscore the survey's significance. Additionally, the survey addresses knowledge gaps in detection techniques and the limitations of signature-based detection in identifying novel and obfuscated malware [1]. The integration of Information Flow Control (IFC) techniques during design and development phases is often overlooked, with existing methods constrained by their inability to adapt to complex application architectures.

This survey provides vital insights into the current cybersecurity landscape, emphasizing the need for innovative approaches to counter the sophisticated and evolving nature of cyber threats [15].

1.4 Structure of the Survey

This survey is meticulously structured to provide a comprehensive examination of adversarial malware and AI security, reflecting the multifaceted nature of these challenges in the contemporary cybersecurity landscape. The paper begins with an **Introduction** section, establishing the significance of adversarial malware and AI security, discussing the motivation for the survey, and highlighting its relevance in the current cybersecurity environment. This is followed by the **Background and Definitions** section, which offers an in-depth overview of essential concepts and terminologies, setting the stage for detailed discussions in subsequent sections.

The third section, **Adversarial Malware and Machine Learning Evasion**, delves into the techniques employed by adversarial malware to evade detection by machine learning models, addressing the challenges these techniques pose to cybersecurity. This is complemented by the **AI-Based Malware Detection Techniques** section, exploring various AI-driven methods designed to enhance malware detection, including image-based and graph-based approaches, as well as novel AI architectures.

The survey then transitions to the **Dynamic Analysis in Malware Detection** section, highlighting the role of dynamic analysis in understanding malware behavior and its integration with static analysis for comprehensive threat detection. Following this, the **Adversarial Attacks on AI Systems** section analyzes the nature of adversarial attacks targeting AI systems, discussing their implications for AI security and illustrating these with case studies and examples.

In the penultimate section, **AI Security and Defense Mechanisms**, the survey reviews strategies and technologies developed to secure AI systems against adversarial attacks, focusing on the development of robust AI models and innovative defense mechanisms. Finally, the **Conclusion** summarizes the key findings of the survey, reflecting on the current state of adversarial malware and AI security, and

suggesting future research directions to address the challenges identified in the paper.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Terminology and Key Concepts

Grasping the terminology and concepts of adversarial malware and AI security is crucial for addressing contemporary cyber threats. Adversarial malware exploits machine learning classifiers by creating adversarial examples—inputs that maintain functionality while deceiving models into benign classification [3]. The generation of these adversarial examples, especially without access to model internals, complicates robust detection system maintenance [12]. Machine learning techniques, including supervised, unsupervised, deep, and reinforcement learning, are pivotal in cybersecurity, yet they remain vulnerable to adversarial tactics [2]. Classifying malware evasion techniques into manual and automated dynamic analysis highlights the need for sophisticated detection models [10]. Dynamic analysis, which observes system behavior during execution, enhances adaptability to new malware by treating reports as documents for text classification [1]. Adversarial evasion attacks manipulate malware features to ensure misclassification, emphasizing the need for robust definitions and models [11].

Key concepts in malware detection include static analysis, which examines code without execution, and dynamic analysis, which involves execution monitoring. Static analysis struggles with dynamically generated code or obfuscation, necessitating dynamic analysis integration for comprehensive threat detection [10]. Polymorphic and metamorphic malware use obfuscation to evade detection, underscoring the need for advanced dynamic analysis methods [1]. In ransomware detection, Adaptive Behavior-Based Ransomware Detection (ABRD) provides real-time behavioral analysis, overcoming signature-based limitations [11].

Static analysis on programs using dynamic code generation obscures actual control flow and data manipulation, necessitating advanced detection techniques [2]. Effective attack graph analysis, representing potential network attack paths, is critical for both static and dynamic risk assessments [18]. In mobile malware detection, binary classification distinguishes malicious from benign applications, highlighting robust feature evaluation's importance in machine learning models [6]. Terms like 'Function Call Graph (FCG)' and 'Graph Convolution Network (GCN)' are crucial in models like Mal2GCN for malware detection [19]. Limited API call representation hampers model performance and generalization [16].

These concepts underpin the development of robust defense mechanisms and enhance AI security against emerging adversarial threats. By integrating machine learning, deep learning, and bio-inspired computing, cybersecurity frameworks aim to elucidate adversarial perturbations' impact on predictions, improving detection models' robustness [12]. Understanding terms like PDF malware, adversarial attacks, and evasion techniques is vital for comprehending adversarial malware and AI security [15]. The benchmark addresses malware detection and classification challenges, particularly in the presence of adversarial attacks that mislead systems [4]. Furthermore, malware classification involves categorizing samples into families based on features, complicated by static analysis's high expertise requirements [5]. Challenges in recovering full DNN model specifications from executables compiled by various compilers further emphasize essential terms and concepts related to adversarial malware and AI security [20].

2.2 Cybersecurity Threats and Adversarial Attacks

The cybersecurity threat landscape is increasingly characterized by sophisticated adversarial attacks exploiting machine learning algorithm vulnerabilities. These attacks mislead AI models, causing malicious software to be misclassified as benign, undermining malware detection systems' integrity. Crafting adversarial examples—inputs engineered to deceive AI models—poses a significant challenge, with their transferability across models complicating robust detection system maintenance [15].

The dynamic nature of evasion attacks further complicates the cybersecurity environment. Malware authors employ advanced techniques to bypass dynamic analysis, crucial for understanding and combating malware. These techniques exploit static and dynamic analysis limitations, particularly

when traditional methods rely on predefined signatures that fail against novel strains [6]. Integrating static and dynamic analysis is essential for comprehensive malware behavior understanding and enhanced detection capabilities.

Implementing effective machine learning solutions for real-time attack detection is complicated by autonomous systems' complexity and data's dynamic nature. Existing benchmarks often use outdated datasets inadequately representing the current Android ecosystem, leading to unreliable malware detection method evaluations [6]. This limitation underscores the need for adaptive detection models responsive to evolving threats.

Moreover, existing security measures' inadequacy during design and development phases emphasizes the need for innovative solutions addressing application architecture complexities and attack vector variety [2]. High false positive rates in heuristic-based methods and signature-based detection limitations illustrate challenges in identifying cybersecurity threats. Research indicates that 41% of machine learning applications do not protect their models, and among those that do, 66% remain vulnerable to extraction through simple techniques, revealing significant model protection vulnerabilities [18].

In recent years, the intersection of cybersecurity and machine learning has garnered significant attention, particularly in the context of adversarial malware. Understanding the complexities involved in this domain is crucial for developing effective countermeasures. Figure 2 illustrates the hierarchical structure of adversarial malware and machine learning evasion, highlighting key techniques, challenges, and vulnerabilities. This figure categorizes machine learning evasion strategies, detection challenges, and AI model vulnerabilities, thereby providing insights into the complex interplay between malware tactics and cybersecurity defenses. Such a comprehensive overview not only elucidates the various dimensions of this issue but also sets the stage for a deeper exploration of the implications for future research and practice in the field.

3 Adversarial Malware and Machine Learning Evasion

3.1 Machine Learning Evasion Techniques

Adversarial malware leverages advanced evasion techniques to bypass machine learning models, exploiting their inherent vulnerabilities. A key tactic involves altering specific bytes in malware binaries, such as those in the DOS header, to evade deep learning classifiers, emphasizing the need for robust detection models [3]. Enhancing resilience against such attacks involves strategies like adversarial training, real-time adaptive defenses, and ensemble methods [9]. Models like Mal2GCN, which utilize Function Call Graphs (FCGs), are developed to counter adversarial exploits targeting deep learning vulnerabilities [19].

Dynamic analysis techniques have evolved, with algorithms like WMRecon reconstructing WM bytecode by identifying primitive behaviors during analysis, offering insights into malware tactics [11]. The ISCCI method enhances evasion rates by dynamically injecting adversarial perturbations into specific sections of a PE file while maintaining functionality [7]. These sophisticated methods challenge traditional detection systems.

Obfuscation techniques further complicate detection, necessitating innovative approaches that combine dynamic and static analysis for comprehensive threat detection [1]. A benchmark study highlights vulnerabilities in mobile applications and the effectiveness of protection methods, underscoring the need for improved security measures [18]. Fine-grained hierarchical learning approaches, such as DL-FHMC, leverage graph mining techniques to enhance detection capabilities [4]. Visual representations, like VisMal, convert binaries into grayscale images to facilitate classification and evade traditional detection methods [5]. These strategies emphasize the necessity for adaptive detection models to respond to the evolving threat landscape.

As illustrated in Figure 3, adversarial malware presents a significant challenge in cybersecurity, particularly regarding machine learning evasion techniques. The first image depicts the malware creation cycle, where a model owner develops a target model that a malware author manipulates to craft a surrogate model, ultimately leading to malware designed to attack unsuspecting users. The second image outlines the architecture of a machine learning model for malware detection, including components like "Malware" input, "Generator," and "Critic," which collaboratively work to identify and mitigate threats. The third image contrasts byte distributions, highlighting the nuanced strategies adversaries employ to evade detection. Collectively, these representations underscore the ongoing

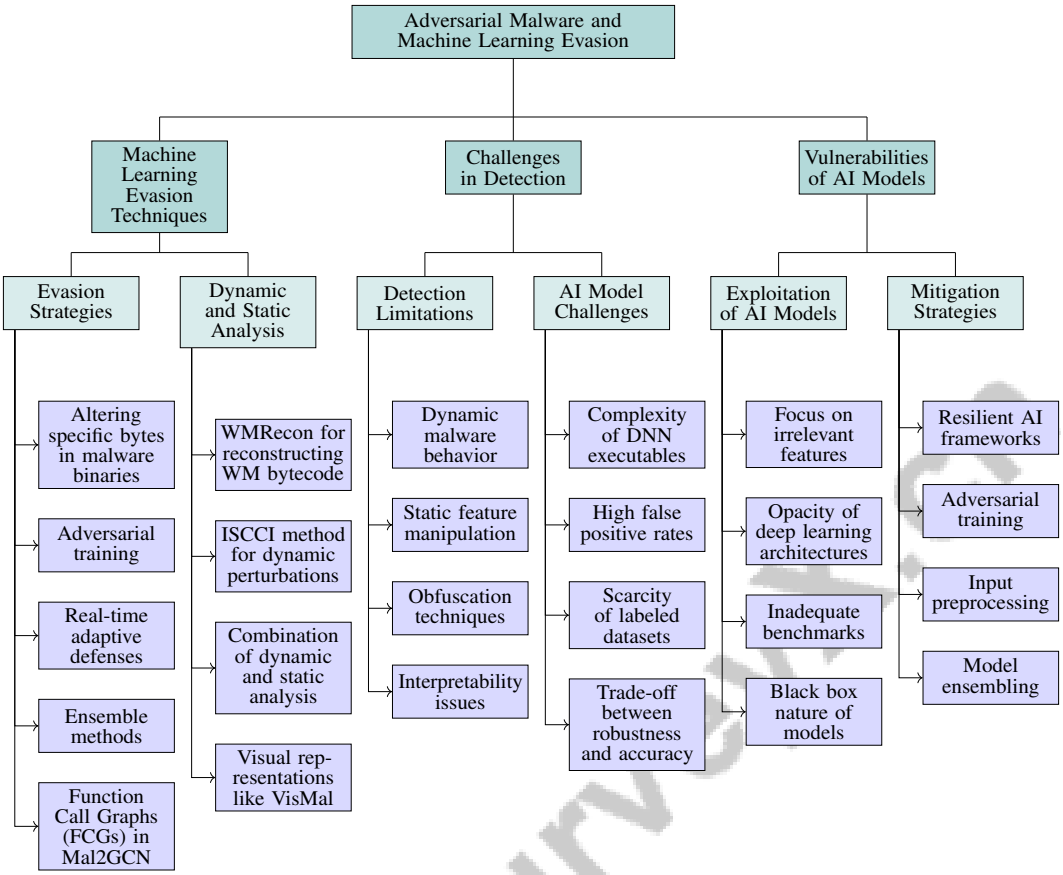
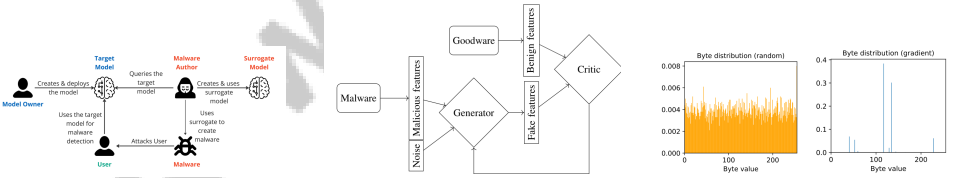


Figure 2: This figure illustrates the hierarchical structure of adversarial malware and machine learning evasion, highlighting key techniques, challenges, and vulnerabilities. It categorizes machine learning evasion strategies, detection challenges, and AI model vulnerabilities, providing insights into the complex interplay between malware tactics and cybersecurity defenses.



(a) Malware Detection and Malware Creation[21]

(b) A diagram illustrating a machine learning model for detecting malware[22]

(c) Byte distribution (random vs. gradient)[23]

Figure 3: Examples of Machine Learning Evasion Techniques

struggle between malware developers and cybersecurity professionals, emphasizing the critical role of advanced machine learning techniques in strengthening defenses against evolving threats [21, 22, 23].

3.2 Challenges in Detection

Detecting adversarial malware presents significant challenges due to sophisticated evasion techniques and the limitations of current detection methodologies. The dynamic nature of malware behavior requires adaptive detection mechanisms. Traditional classifiers often struggle with robustness against adversarial tactics, as they rely on static features that adversarial malware can manipulate [15]. Existing static and dynamic analysis methods fail to adapt to rapidly changing malware signatures, exacerbated by obfuscation techniques that alter static characteristics, rendering conventional methods

ineffective [1]. Interpretability issues with deep learning models, often operating as black boxes, further complicate detection by obscuring the features driving classification decisions [3].

Recovering full specifications from deep neural network (DNN) executables presents another challenge. The complexity of DNNs, particularly those compiled by various deep learning compilers, complicates understanding and mitigation of adversarial threats [20]. This highlights the need for advanced techniques to effectively decompile and analyze DNN executables. AI systems also face challenges related to high false positive rates and the requirement for extensive labeled datasets to train models effectively. The scarcity of such datasets hampers the development of robust models, as acquiring comprehensive training data is not always feasible [2]. The trade-off between robustness and detection accuracy remains a concern; adversarial training methods, while enhancing robustness, may decrease performance on clean data [9].

3.3 Vulnerabilities of AI Models

AI model vulnerabilities in adversarial malware detection stem from inherent weaknesses in deep learning frameworks. Models like MalConv often focus on irrelevant features, which adversarial malware exploits to evade detection [3]. This focus on non-essential features makes models susceptible to adversarial inputs crafted to mislead AI into misclassifying malicious software as benign. Addressing these vulnerabilities is complicated by inadequate benchmarks that fail to evaluate adversarial malware effectiveness against current antivirus software, highlighting deficiencies in detection models not rigorously tested against sophisticated tactics [24].

The exploitation of AI model vulnerabilities by adversarial malware is further exacerbated by the complexity and opacity of deep learning architectures. These models function as black boxes, making it difficult to interpret decision-making processes and identify specific features manipulated by adversarial inputs. This opacity complicates efforts to fortify models against adversarial attacks and hinders the development of more transparent and interpretable AI systems [3]. To mitigate vulnerabilities, there is an urgent need for resilient AI frameworks incorporating adaptive defense mechanisms like adversarial training, input preprocessing, and model ensembling, dynamically evolving to counteract sophisticated evasion attacks while ensuring accuracy and efficiency across applications, including image recognition and malware detection. By proactively adjusting to emerging threats and enhancing interpretability, these robust frameworks can bolster AI security and reliability [25, 26, 27, 9, 28]. This includes integrating adversarial training methods and refining model architectures to focus on relevant features. By addressing shortcomings in evaluation benchmarks and enhancing interpretability, the cybersecurity community can better prepare for sophisticated adversarial tactics employed by modern malware.

4 AI-Based Malware Detection Techniques

4.1 AI-Based Malware Detection

The integration of artificial intelligence (AI) into malware detection marks a significant advancement in identifying and mitigating cyber threats. Techniques such as convolutional neural networks (CNNs) and deep neural networks (DNNs) are pivotal in enhancing detection capabilities. For instance, the VisMal framework employs a CNN to achieve a 96.0% accuracy in classifying malware, demonstrating the efficacy of CNNs when integrated with innovative frameworks [5]. Random Forest (RF) models also exhibit strong performance, with a benchmark study revealing a 98.90% accuracy, highlighting the robustness of ensemble methods in handling diverse malware datasets [4]. These findings underscore the importance of combining multiple learning paradigms to improve detection accuracy and resilience against adversarial strategies.

Machine learning classifiers' vulnerabilities to adversarial tactics, especially in PDF malware detection, necessitate robust frameworks to withstand such manipulations [15]. Explainable AI (XAI) methods enhance transparency and trust in AI systems by elucidating deep learning models' decision-making processes [3]. The BTD (Bin to DNN) method significantly advances AI-based malware detection by decompiling deep neural networks from executables, enhancing model interpretability and understanding of their vulnerabilities [20].

4.2 Image-Based Malware Detection

Image-based techniques offer a novel approach to malware detection by converting malware binaries into visual representations, such as grayscale images, analyzed using computer vision techniques. This method uncovers nuanced patterns and features often overlooked by traditional approaches, particularly through advanced machine learning classifiers applied to extracted features from malware binary images [27, 29]. The VisMal framework exemplifies this by using CNNs to classify these visual representations, achieving high classification accuracy [5]. Transforming malware into images enables the use of advanced image processing techniques, such as hybrid fuzzing, as illustrated by datasets in [30], which can be integrated into detection frameworks to identify new malware variants.

Integrating image-based techniques with existing machine learning models creates a comprehensive framework for enhancing malware detection. This approach leverages feature extraction from malware binary images, improving malware class differentiation and addressing challenges posed by stealthy and adversarial attacks. By combining visual analysis with traditional binary analysis, cybersecurity professionals can develop more resilient detection systems capable of adapting to evolving threats [31, 27, 17, 29].

4.3 Graph-Based Detection Techniques

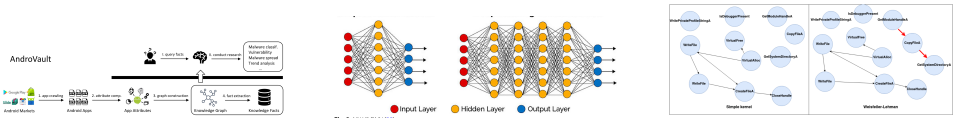
Method Name	Structural Features	Detection Techniques	Integration Methods
M2GCN[19]	Function Call Graph	Graph Convolution Network	Dynamic And Static
WL-Kernel[32]	System Call Dependency	Support Vector Machines	Symbolic Execution

Table 1: Table 1 presents a comparative analysis of two graph-based malware detection methods, M2GCN and WL-Kernel, highlighting their structural features, detection techniques, and integration methods. The table emphasizes the use of function call graphs and system call dependencies in enhancing adversarial malware detection and symbolic execution, respectively. This comparison underscores the integration of dynamic and static analysis in improving detection accuracy and resilience against evolving threats.

Graph-based detection techniques offer unique insights into the structural and behavioral patterns of malware by utilizing software's inherent graph-like structure. Nodes represent entities such as functions or system calls, while edges denote interactions or data flows. Analyzing these graphs uncovers intricate relationships and dependencies within malware behaviors, enhancing detection and classification accuracy. Advanced techniques, like graph attention networks and deep learning, identify patterns in malware activity, particularly during dynamic analysis, improving real-time detection and contributing to resilient cybersecurity frameworks [33, 31, 34, 27, 35].

Function Call Graphs (FCGs) are notably applied in malware analysis. The Mal2GCN model uses graph convolutional networks (GCNs) to process FCGs, enhancing adversarial malware detection by focusing on code's structural properties [19]. Graph-based techniques also integrate dynamic and static analysis methods, treating dynamic analysis reports as graph-like documents to identify novel malware variants and predict attack paths [1]. This integration is crucial for addressing traditional detection methods' limitations in adapting to evolving threats.

Graph-based techniques extend to large-scale network analysis, where attack graphs represent potential attack paths through a system, vital for static and dynamic risk assessments [18]. By visualizing network component interconnections, these techniques provide valuable security posture insights and help identify critical vulnerabilities.



(a) AndroVault: A System for Malware Classification and Vulnerability Analysis[36]

(b) Comparison of Neural Network (NN) and Deep Neural Network (DNN) Architectures[27]

(c) Comparison of Kernel Implementations[32]

Figure 4: Examples of Graph-Based Detection Techniques

As illustrated in Figure 4, graph-based detection techniques are formidable tools in AI-based malware detection. AndroVault employs graph construction and fact extraction to classify malware and assess vulnerabilities in Android applications, constructing knowledge graphs that provide insights into malware behavior and trends. The comparison of neural network (NN) and deep neural network (DNN) architectures highlights DNNs' increased complexity and interconnectivity, enhancing AI model detection capabilities. The comparison of kernel implementations illustrates execution flow differences between "Simple kernel" and "Weisfeiler-Lehman" methods, emphasizing the importance of selecting appropriate techniques for effective malware analysis [36, 27, 32]. Furthermore, Table 1 provides a detailed comparison of graph-based detection techniques, illustrating the structural features, detection methods, and integration approaches employed by the M2GCN and WL-Kernel models.

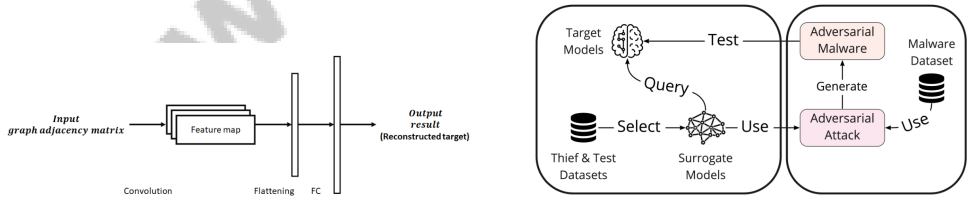
4.4 Novel AI Architectures for Malware Detection

Method Name	Architectural Approach	Data Representation	Detection Enhancement
M2GCN[19]	Graph Convolution Network	Function Call Graphs	Adversarial Detection
VM[5]	Convolutional Neural Network	Grayscale Images	Contrast Enhancement
BTD[20]	Systematic Decompilation Approach	Representation Learning	Dynamic Analysis
CNN[37]	Fusion Neural Network	Composite Neural Network	Feature Integration
DFNN[21]	Dual Architecture	True Labels	Adversarial Detection

Table 2: Comparison of novel AI-based malware detection methods, detailing their architectural approaches, data representation techniques, and detection enhancement strategies. The table highlights the diversity of methods employed, ranging from graph convolution networks to convolutional neural networks, and their corresponding data handling and enhancement mechanisms.

Developing novel AI architectures for malware detection enhances cybersecurity measures' efficacy and robustness. These architectures utilize advanced machine learning techniques to detect and mitigate sophisticated cyber threats that traditional methods struggle to address. The Mal2GCN model employs a Graph Convolution Network (GCN) to process Function Call Graphs (FCGs) for robust malware detection, highlighting graph-based models' potential in identifying complex patterns within malicious software [19]. The DL-FHMC model employs a fine-grained hierarchical learning approach to enhance adversarial detection capabilities, leveraging graph mining techniques for a robust framework that adapts to evolving threats [4].

The VisMal framework showcases innovative AI architectures' potential in malware detection by transforming malware binaries into grayscale images and utilizing CNNs for classification, achieving high accuracy [5]. The BTD (Bin to DNN) method advances understanding AI-based malware detection by decompiling deep neural networks from executables, enhancing model interpretability and understanding of their functionality and vulnerabilities [20].



(a) Convolutional Neural Network for Graph-Structured Data[37]

(b) Adversarial Malware Detection System[21]

Figure 5: Examples of Novel AI Architectures for Malware Detection

As shown in Figure 5, exploration of AI-based malware detection techniques has led to novel architectures leveraging advanced machine learning models to enhance security measures. The Convolutional Neural Network (CNN) for Graph-Structured Data processes complex relationships within graph data, allowing nuanced analysis of malware patterns. The Adversarial Malware Detection System employs a dual-model strategy, comprising a target and a surrogate model, to generate and detect adversarial malware, enhancing detection capabilities by simulating attacks and analyzing outcomes. These architectures represent cutting-edge advancements in AI-driven cybersecurity, offering robust frameworks to counteract evolving malware threats [37, 21]. Table 2 provides a

AI-generated, for reference only.

comprehensive comparison of various novel AI architectures developed for malware detection, illustrating the diverse methodologies and techniques employed to enhance cybersecurity measures.

5 Dynamic Analysis in Malware Detection

5.1 The Role of Dynamic Analysis in Malware Detection

Dynamic analysis is vital for malware detection, offering insights into runtime behaviors that static analysis misses. By executing malware in controlled settings, it reveals real-time interactions crucial for identifying sophisticated threats that evade traditional methods [1]. It is particularly effective against polymorphic and metamorphic variants, capturing dynamic features like API calls and resource consumption, thereby providing a comprehensive view of malware activities [6]. The synergy between dynamic and static analysis enhances detection by extracting more complete features, improving accuracy and resilience against evasion strategies [4]. For instance, WMRecon utilizes dynamic analysis to detect exploit behaviors, deepening understanding through bytecode reconstruction [11].

Despite its strengths, dynamic analysis faces challenges like processing time and complexity. The VisMal framework addresses these by employing efficient image-based classification to streamline analysis while maintaining accuracy [5]. Additionally, dynamic analysis is essential for assessing machine learning models' vulnerabilities, highlighting their susceptibility to adversarial attacks and the need for robust detection frameworks [20].

5.2 Integration of Static and Dynamic Analysis

Integrating static and dynamic analysis offers a comprehensive approach to malware detection, combining their strengths to improve accuracy and resilience against sophisticated threats. Static analysis assesses code structure without execution, identifying vulnerabilities and malicious patterns but often struggles with obfuscated code and dynamic features. Dynamic analysis captures actual malware behavior during execution, monitoring interactions like process creation and network connections, crucial for understanding malware behavior [38]. A composite neural network integrating features from both analyses demonstrates hybrid approaches' potential to enhance detection [37]. Experiments on Android applications validate this integration, where hybrid approaches outperform methods relying solely on random or state-based techniques, underscoring the effectiveness of combining static and dynamic data [39].

Dynamic analysis optimization can be achieved using techniques like term frequency–inverse document frequency (TF-IDF) for weight calculation and sliding window algorithms for data preprocessing, facilitating efficient and accurate detection [40]. Platforms like Andlantis exemplify dynamic analysis scalability, processing over 3,000 Android applications per hour, demonstrating the viability of large-scale analysis through parallel execution [41]. Integrating static and dynamic analysis is crucial for enhancing Android application malware detection, effectively addressing each approach's limitations when used alone [42]. By combining static code analysis with dynamic runtime behavior observation, detection models can be trained on diverse inputs, improving accuracy in identifying and classifying malware [43]. Future research should focus on expanding analyses supported by tools like DynaPyt and addressing existing limitations to further enhance detection [44].

5.3 Challenges and Limitations of Dynamic Analysis

Dynamic analysis, despite its critical role in malware detection, faces challenges that can hinder its effectiveness. A key issue is the imbalanced contribution of concatenated static and dynamic feature vectors, complicating integration and reducing detection efficacy [45]. This imbalance may lead to incomplete threat detection when combining dynamic analysis with static methods. Another challenge is the inadequate exercising of the user interface (UI), crucial for uncovering specific malicious behaviors. Many tools fail to engage comprehensively with the UI, potentially missing critical actions by malware, exacerbated by naive UI exploration algorithms in large-scale Android dynamic analysis systems [41].

The hybrid dynamic analysis approach, merging state-based and random methods, also presents challenges. State-based approaches, like DroidBot, can be slower than random methods like Monkey due to the overhead of static analysis and environment setup, delaying detection [39]. Furthermore,

platforms like Glassbox may be vulnerable to fingerprinting by malware, allowing it to detect the analysis environment and alter its behavior to evade detection [46]. Handling JavaScript code within dynamic analysis poses another challenge. Tools like JSForce may terminate analysis if JavaScript syntax is incorrect, missing malicious script detections [47]. Additionally, logging outputs from native code complicates analysis, as seen with DynaLog, where evasion techniques further complicate detection [48].

The evolving nature of ransomware variants introduces additional challenges, as reliance on specific datasets and biases in training data may allow sophisticated ransomware to evade detection [49]. Tools like JITScanner, which depend on dynamic executable page nature, may fail to capture all malicious activities if the code does not trigger instruction fetches, leaving some threats undetected [50]. Dynamic analysis is also constrained by computational expenses and increased complexity in generating adversarial examples, particularly in high-dimensional feature spaces [51]. The method may not uncover all vulnerabilities, especially those requiring specific conditions to trigger, limiting its comprehensiveness [13]. Additionally, the ISCCI method's reliance on specific section sizes within malware files may restrict its applicability across all samples [7].

Finally, the lack of a comprehensive framework outlining diverse strategies and implementations of dynamic analysis tools complicates the selection process for analysts, who may struggle to choose the optimal tool for their needs [52]. Addressing these challenges is essential for enhancing dynamic analysis effectiveness and ensuring comprehensive threat detection in the evolving malware threat landscape.

6 Adversarial Attacks on AI Systems

6.1 Nature and Techniques of Adversarial Attacks

Adversarial attacks exploit vulnerabilities in machine learning models, allowing crafted inputs to misclassify malicious entities as benign. These attacks, categorized into evasion and poisoning types, aim to bypass detection mechanisms and compromise model integrity [15]. Evasion attacks generate adversarial examples that maintain malware functionality while avoiding detection, with combined adversarial generators enhancing their effectiveness against antivirus solutions [19]. Deep learning models for malware detection are particularly vulnerable to adversarial attacks, which manipulate models into misclassification, threatening system reliability [53]. New algorithms demonstrate precision by altering minimal bytes to generate adversarial malware, contrasting with traditional extensive modifications [3]. These attacks also impact IoT networks, significantly degrading detection accuracy and highlighting the need for robust defenses [54]. The PAD framework exemplifies strategic attack combinations to enhance robustness, illustrating diverse adversarial tactics [12]. Challenges in malware classification due to evasion techniques necessitate innovative detection frameworks to counter sophisticated threats [1]. Visual malware representations, as in frameworks like VisMal, may improve detection by addressing traditional weaknesses [5].

6.2 Implications for AI Security

Adversarial attacks significantly challenge AI security in malware detection by exploiting vulnerabilities to mislead models into incorrect classifications. These threats demand robust defenses to protect AI systems from sophisticated manipulations [2]. The adaptive nature of adversarial tactics, especially regarding ransomware, necessitates evolving defense mechanisms. Recent experiments emphasize benchmarks' value in revealing vulnerabilities of machine learning-based detection systems, providing insights for enhancing robustness against adversarial attacks [4]. Challenges are compounded by potential recovery errors from obfuscated executables, highlighting the need to address obfuscation techniques and ensure accurate AI model interpretation [20]. Integrating dynamic analysis into detection strategies is crucial for improving generalization to unseen malware, as static features alone may be insufficient for real-time detection. Dynamic features enhance resilience against adversarial tactics, though real-time detection complexity remains significant, requiring further validation across diverse malware types [55]. Adversarial attacks also necessitate innovative methodologies to tackle obfuscation and automated malware generation issues, complicating detection. Advanced representation generation methods can improve performance while addressing weak generalization and concept drift [56]. The risk of model leakage in mobile applications underscores the need for robust protection mechanisms to safeguard AI systems against adversarial threats [18].

6.3 Case Studies and Examples

Case studies vividly illustrate adversarial attacks' impact on AI systems, highlighting vulnerabilities and challenges in current detection methodologies. One study reveals AI-driven malware detection systems' susceptibility to adversarial obfuscation techniques designed to conceal malicious flows, undermining AI model integrity. Semantic analysis of sensitive data flows in Android applications demonstrates vulnerability to such obfuscation, revealing significant limitations in detection frameworks [57]. In Android malware detection, ongoing adversarial tactics evolution necessitates detection benchmarks' enhancement. Future research aims to expand datasets and integrate additional feature extraction tools, improving robustness against adversarial attacks [58]. These case studies underscore the need for innovative solutions and adaptive frameworks capable of dynamically responding to sophisticated adversarial strategies, particularly in evasion attacks on AI models. The analysis highlights adaptive defense mechanisms' effectiveness, such as adversarial training and input preprocessing, enhancing model resilience and maintaining accuracy against various attack methodologies [27, 9]. By examining real-world examples, researchers and practitioners gain deeper insights into adversarial attacks' complexities, developing more resilient AI systems to maintain robust defenses in an evolving threat landscape.

7 AI Security and Defense Mechanisms

7.1 Development of Robust AI Models

Enhancing the robustness of AI models is crucial for defending against sophisticated adversarial attacks on digital infrastructures. Adaptive defense mechanisms, which evolve with the threat landscape, are vital for maintaining AI resilience against both known and unknown adversarial threats [9]. Implementing non-negative weight constraints, as demonstrated by the Mal2GCN model, improves resilience by leveraging malware code's structural properties, thereby enhancing detection capabilities [19]. The automation of vulnerability detection and prioritization through methods like Sydr-Fuzz emphasizes the need for prompt responses to security issues, ensuring robust AI defenses [13].

Future research should focus on developing robust protection mechanisms for machine learning models and exploring secure hardware solutions for on-device ML [18]. Adversarial training, which incorporates adversarial examples into the training process, is a pivotal strategy for fortifying defenses and enhancing model generalization across diverse threat scenarios [15].

7.2 Innovative Defense Mechanisms

Innovative defense mechanisms are essential for safeguarding AI systems from sophisticated adversarial attacks. The MalFox framework illustrates the need for advanced strategies to counter adversarial malware that evades traditional detection systems [59]. Adversarial training has proven effective in enhancing AI model resilience, allowing models to resist manipulative inputs by including adversarial examples during training [60]. This approach reduces misclassification risks and enhances robustness across various threat scenarios.

Adaptive defense mechanisms are critical for maintaining AI efficacy against dynamic threats. By integrating strategies such as adversarial training, defensive distillation, input preprocessing, and model ensembling, AI frameworks can dynamically respond to evolving attack methodologies, thus enhancing resilience [27, 2, 9, 61]. Exploring novel architectures and algorithms that focus on malware's structural properties further strengthens defense mechanisms, particularly through advanced graph-based models and innovative training methodologies [27, 2].

7.3 Defense Mechanisms Against Adversarial Attacks

Effective defense mechanisms are crucial for maintaining AI systems' integrity and reliability in cybersecurity. Adversarial training, which incorporates adversarial examples during training, enhances models' resilience to attacks and improves generalization across diverse threat scenarios [60]. Adaptive defense strategies facilitate the continuous evolution of detection frameworks, enabling AI models to adapt to the dynamic nature of adversarial threats. Strategies such as adversarial training, input preprocessing, and model ensembling empower AI systems to proactively counteract evasion

attacks, thus enhancing robustness and fostering trust through transparency and interpretability [27, 9, 28].

Exploring novel architectures and algorithms that emphasize malware's structural properties can lead to more robust defense mechanisms. Advanced graph-based models and innovative training techniques enhance AI systems' detection capabilities, equipping them to handle adversarial attacks' complexities [19]. Ensemble methods, which combine multiple learning paradigms, show promise in improving detection accuracy and resilience against adversarial strategies, creating a more robust detection framework [4].

8 Conclusion

8.1 Future Research Directions

Advancing research in adversarial malware and AI security necessitates the development of adaptive models that can dynamically learn and respond to new threats. Enhancing the interpretability of machine learning models is crucial for understanding adversarial attacks and fortifying defenses. Integrating AI with human oversight and exploring bio-inspired computation methods holds potential for strengthening cybersecurity. Investigating black-box scenarios and vulnerabilities within PE file structures is essential for broadening our understanding of adversarial tactics and improving detection methodologies. Moreover, crafting sophisticated attacks that evade detection will shed light on vulnerabilities in deep learning applications beyond malware detection.

Automating data collection and expanding datasets are vital for enhancing the robustness of dynamic analysis against evolving malware threats. Future research should also focus on validating dynamically growing knowledge graphs for predicting threat vectors and improving threat intelligence. Enhancing the capability of frameworks like VisMal to differentiate between malware and benign software is key to improving classification performance. Exploring principled methods for adversarial training and addressing functionality preservation in adversarial examples remains critical for advancing AI security.

Further, expanding tools to support additional DNN operators and improving robustness against obfuscation and advanced compiler optimizations is necessary to address adversarial malware challenges. Investigating complex exploit scenarios involving multiple scripting languages will provide deeper insights into modern cyber threats. Pursuing these research avenues will enable the development of more resilient solutions against evolving adversarial malware threats, ultimately enhancing the security of AI-driven systems in an increasingly complex threat landscape.

8.2 Advancements and Future Directions

Recent advancements underscore the effectiveness of integrating dynamic analysis with machine learning and deep learning techniques, particularly in ransomware detection. This integration has proven effective in identifying and mitigating ransomware threats, underscoring the importance of combining traditional analysis methods with advanced AI-driven approaches. The dynamic nature of ransomware necessitates adaptive detection mechanisms capable of responding to evolving threat patterns, marking a critical area for future research.

Employing dynamic programming-based approaches to generate adversarial payloads has shown promise in evading detection systems, highlighting the need for ongoing research into adaptive defense strategies. Additionally, understanding the motivations behind AI-driven cyberattacks and their societal impacts is crucial for developing comprehensive strategies that enhance technical defenses while considering the social and ethical implications of AI in cybersecurity.

The continuous evolution of adversarial techniques and increasing sophistication of cyber threats necessitate reevaluating existing methodologies and exploring novel approaches. Future research should prioritize developing robust AI models that can adapt to new threats, ensuring the resilience and reliability of cybersecurity systems. By addressing these advancements and existing research gaps, the cybersecurity community can improve its capacity to protect against the multifaceted challenges posed by adversarial malware and AI-driven cyberattacks.

References

[1] Cengiz Acarturk, Melih Sirlanci, Pinar Gurkan Balikcioglu, Deniz Demirci, Nazenin Sahin, and Ozge Acar Kucuk. Malicious code detection: Run trace output analysis by lstm, 2021.

[2] Thanh Cong Truong, Quoc Bao Diep, and Ivan Zelinka. Artificial intelligence in the cyber domain: Offense and defense. *Symmetry*, 12(3):410, 2020.

[3] Luca Demetrio, Battista Biggio, Giovanni Lagorio, Fabio Roli, and Alessandro Armando. Explaining vulnerabilities of deep learning to adversarial malware binaries, 2019.

[4] Ahmed Abusnaina. Studying the robustness of machine learning-based malware detection models: Analysis, design, and implementation. 2022.

[5] Fangtian Zhong, Zekai Chen, Minghui Xu, Guoming Zhang, Dongxiao Yu, and Xiuzhen Cheng. Malware-on-the-brain: Illuminating malware byte codes with images for malware classification, 2022.

[6] Ryan Christian, Sharmishtha Dutta, Youngja Park, and Nidhi Rastogi. Ontology-driven knowledge graph for android malware, 2021.

[7] Kshitiz Aryal, Maanak Gupta, Mahmoud Abdelsalam, and Moustafa Saleh. Intra-section code cave injection for adversarial evasion attacks on windows pe malware file, 2024.

[8] Xin Wang and Siu Ming Yiu. A multi-task learning model for malware classification with useful file access pattern from api call sequence, 2016.

[9] Bilal Shah. Adaptive defense strategies for protecting ai models from evasion attacks in adversarial machine learning. *Aitoz Multidisciplinary Review*, 3(1):323–337, 2024.

[10] Ning Xi, Chao Chen, Jun Zhang, Cong Sun, Shigang Liu, Pengbin Feng, and Jianfeng Ma. Information flow based defensive chain for data leakage detection and prevention: a survey, 2021.

[11] Robert Abela and Mark Vella. Casting exploit analysis as a weird machine reconstruction problem, 2021.

[12] Deqiang Li, Shicheng Cui, Yun Li, Jia Xu, Fu Xiao, and Shouhuai Xu. Pad: Towards principled adversarial malware detection against evasion attacks, 2023.

[13] Ilya Yegorov, Eli Kobrin, Darya Parygina, Alexey Vishnyakov, and Andrey Fedotov. Python fuzzing for trustworthy machine learning frameworks, 2024.

[14] Lorenzo Maffia, Dario Nisi, Platon Kotzias, Giovanni Lagorio, Simone Aonzo, and Davide Balzarotti. Longitudinal study of the prevalence of malware evasive techniques, 2021.

[15] Davide Maiorca, Battista Biggio, and Giorgio Giacinto. Towards adversarial malware detection: Lessons learned from pdf-based attacks, 2020.

[16] Savino Dambra, Yufei Han, Simone Aonzo, Platon Kotzias, Antonino Vitale, Juan Caballero, Davide Balzarotti, and Leyla Bilge. Decoding the secrets of machine learning in malware classification: A deep dive into datasets, feature extraction, and model performance, 2023.

[17] Matthew Crawford, Wei Wang, Ruoxi Sun, and Minhui Xue. Statically detecting adversarial malware through randomised chaining, 2021.

[18] Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove. Mind your weight(s): A large-scale study on insufficient machine learning model protection in mobile apps, 2021.

[19] Omid Kargarnovin, Amir Mahdi Sadeghzadeh, and Rasool Jalili. Mal2gcn: A robust malware detection approach using deep graph convolutional networks with non-negative weights, 2022.

[20] Zhibo Liu, Yuanyuan Yuan, Shuai Wang, Xiaofei Xie, and Lei Ma. Decompiling x86 deep neural network executables, 2022.

[21] Maria Rigaki and Sebastian Garcia. Stealing and evading malware classifiers and antivirus at low false positive conditions, 2023.

[22] Daniel Gibert, Jordi Planes, Quan Le, and Giulio Zizzo. Query-free evasion attacks against machine learning-based malware detectors with generative adversarial networks, 2023.

[23] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. Adversarial malware binaries: Evading deep learning for malware detection in executables, 2018.

[24] Matouš Kozák and Martin Jureček. Combining generators of adversarial malware examples to increase evasion rate, 2023.

[25] Deqiang Li, Qianmu Li, Yanfang Ye, and Shouhuai Xu. Enhancing robustness of deep neural networks against adversarial malware samples: Principles, framework, and aics'2019 challenge, 2020.

[26] Deqiang Li, Qianmu Li, Yanfang Ye, and Shouhuai Xu. A framework for enhancing deep neural networks against adversarial malware, 2021.

[27] Aya H Salem, Safaa M Azzam, OE Emam, and Amr A Abohany. Advancing cybersecurity: a comprehensive review of ai-driven detection techniques. *Journal of Big Data*, 11(1):105, 2024.

[28] Md Tarek Hossain, Rumi Afrin, and Mohd Al-Amin Biswas. A review on attacks against artificial intelligence (ai) and their defence image recognition and generation machine learning, artificial intelligence. *Control Systems and Optimization Letters*, 2(1):52–59, 2024.

[29] Abhijitt Dhavlle and Sanket Shukla. A novel malware detection mechanism based on features extracted from converted malware binary images, 2021.

[30] Alexey Vishnyakov, Daniil Kuts, Vlada Logunova, Darya Parygina, Eli Kobrin, Georgy Savidov, and Andrey Fedotov. Sydr-fuzz: Continuous hybrid fuzzing and dynamic analysis for security development lifecycle, 2023.

[31] Amir Djenna, Ahmed Bouridane, Saddaf Rubab, and Ibrahim Moussa Marou. Artificial intelligence-based malware detection, analysis, and mitigation. *Symmetry*, 15(3):677, 2023.

[32] Charles-Henry Bertrand Van Ouytsel and Axel Legay. Malware analysis with symbolic execution and graph kernel, 2022.

[33] S. W. Hsiao and P. Y. Chu. Sequence feature extraction for malware family analysis via graph neural network, 2022.

[34] Qian Chen, Sheikh Rabiul Islam, Henry Haswell, and Robert A. Bridges. Automated ransomware behavior analysis: Pattern extraction and early detection, 2019.

[35] Minh Tu Nguyen, Viet Hung Nguyen, and Nathan Shone. Using deep graph learning to improve dynamic analysis-based malware detection in pe files. *Journal of Computer Virology and Hacking Techniques*, 20(1):153–172, 2024.

[36] Guozhu Meng, Yinxing Xue, Jing Kai Siow, Ting Su, Annamalai Narayanan, and Yang Liu. Androvault: Constructing knowledge graph from millions of android apps for automated analysis, 2017.

[37] Yao Saint Yen, Zhe Wei Chen, Ying Ren Guo, and Meng Chang Chen. Integration of static and dynamic analysis for malware family classification with composite neural network, 2019.

[38] Baskoro Adi Pratomo, Toby Jackson, Pete Burnap, Andrew Hood, and Eirini Anthi. Enhancing enterprise network security: Comparing machine-level and process-level analysis for dynamic malware detection, 2023.

[39] Mohammed K. Alzaylaee, Suleiman Y. Yerima, and Sakir Sezer. Improving dynamic analysis of android apps using hybrid test input generation, 2017.

[40] Mihui Kim and Haesoo Kim. A dynamic analysis data preprocessing technique for malicious code detection with tf-idf and sliding windows. *Electronics*, 13(5):963, 2024.

[41] Michael Bierma, Eric Gustafson, Jeremy Erickson, David Fritz, and Yung Ryn Choe. Andlantis: Large-scale android dynamic analysis, 2014.

[42] Francisco Handrick da Costa, Ismael Medeiros, Thales Menezes, João Victor da Silva, Ingrid Lorraine da Silva, Rodrigo Bonifácio, Krishna Narasimhan, and Márcio Ribeiro. Exploring the use of static and dynamic analysis to improve the performance of the mining sandbox approach for android malware identification, 2021.

[43] Anusha Damodaran, Fabio Di Troia, Corrado Aaron Visaggio, Thomas H Austin, and Mark Stamp. A comparison of static, dynamic, and hybrid analysis for malware detection. *Journal of Computer Virology and Hacking Techniques*, 13:1–12, 2017.

[44] Aryaz Eghbali and Michael Pradel. Dynapyt: a dynamic analysis framework for python. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 760–771, 2022.

[45] Mao V. Ngo, Tram Truong-Huu, Dima Rabadi, Jia Yi Loo, and Sin G. Teo. Fast and efficient malware detection with joint static and dynamic features through transfer learning, 2022.

[46] Paul Irolla and Eric Filiol. Glassbox: Dynamic analysis platform for malware android applications on real devices, 2016.

[47] Xunchao Hu, Yao Cheng, Yue Duan, Andrew Henderson, and Heng Yin. Jsforce: A forced execution engine for malicious javascript detection, 2017.

[48] Mohammed K. Alzaylaee, Suleiman Y. Yerima, and Sakir Sezer. Dynalog: An automated dynamic analysis framework for characterizing android applications, 2016.

[49] Umara Urooj, Bander Ali Saleh Al-Rimy, Anazida Zainal, Fuad A Ghaleb, and Murad A Rassam. Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, 12(1):172, 2021.

[50] Pasquale Caporaso, Giuseppe Bianchi, and Francesco Quaglia. Jitscanner: Just-in-time executable page check in the linux operating system, 2024.

[51] Abdullah Al-Dujaili, Alex Huang, Erik Hemberg, and Una-May O'Reilly. Adversarial deep learning for robust detection of binary encoded malware, 2018.

[52] Waqas Aman. A framework for analysis and comparison of dynamic malware analysis tools, 2014.

[53] Marjan Golmaryami, Rahim Taheri, Zahra Pooranian, Mohammad Shojafar, and Pei Xiao. Setti: A self-supervised adversarial malware detection architecture in an iot environment, 2022.

[54] Luca Massarelli, Leonardo Aniello, Claudio Ciccotelli, Leonardo Querzoni, Daniele Ucci, and Roberto Baldoni. Android malware family classification based on resource consumption over time, 2017.

[55] Chenzhong Yin, Hantang Zhang, Mingxi Cheng, Xiongye Xiao, Xinghe Chen, Xin Ren, and Paul Bogdan. Discovering malicious signatures in software from structural interactions, 2023.

[56] Pei Yan, Shunquan Tan, Miaohui Wang, and Jiwu Huang. Prompt engineering-assisted malware dynamic analysis using gpt-4, 2023.

[57] Hao Fu, Zizhan Zheng, Somdutta Bose, Matt Bishop, and Prasant Mohapatra. Leaksemantic: Identifying abnormal sensitive network transmissions in mobile applications, 2017.

[58] Alejandro Martín, Raúl Lara-Cabrera, and David Camacho. Android malware detection through hybrid features fusion and ensemble classifiers: The andropytool framework and the omnidroid dataset. *Information Fusion*, 52:128–142, 2019.

[59] Fangtian Zhong, Xiuzhen Cheng, Dongxiao Yu, Bei Gong, Shuaiwen Song, and Jiguo Yu. Malfox: Camouflaged adversarial malware example generation based on conv-gans against black-box detectors, 2022.

[60] Jacopo Cortellazzi, Feargus Pendlebury, Daniel Arp, Erwin Quiring, Fabio Pierazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ml attacks in the problem space [extended version], 2024.

[61] Masike Malatji and Alaa Tolah. Artificial intelligence (ai) cybersecurity dimensions: a comprehensive framework for understanding adversarial and offensive ai. *AI and Ethics*, pages 1–28, 2024.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.