AI-Enhanced Malware Analysis: Breaking Down Advanced Cyber Threats with Precision

Ahmed Elgalb, Independent Researcher, Iowa, United States

Mahmoud Mohamed, Independent Researcher, Giza, Egypt

Abstract

The rise in global cyber-attacks highlights the need for more sophisticated malware analysis tools and methodologies. As attackers use more advanced techniques, static signatures and heuristic rules are not adequate to detect attacks. The rise of artificial intelligence (AI), including machine learning (ML), deep learning (DL), and anomaly detection, has radically changed the way malware is detected, allowing it to have more adaptive and powerful protection. The paper provides an overview of AI-powered malware analysis — from evolving threats, through basic static and dynamic analysis, to anomaly detection for real-time threat monitoring. As a comparative analysis of AI's effectiveness at detecting polymorphic, metamorphic and zero-day attacks shows, AI technologies are more effective than traditional signature-based approaches. In addition, issues of adversarial machine learning, model interpretability, and data-based retraining pipelines are discussed, mirroring current debates in industry and academia. It ends by identifying the importance of proactive AI systems in contemporary cybersecurity, and suggests research avenues such as federated learning, explainable AI, and aligning regulatory expectations with cutting-edge security.

1. Introduction

Malware, a term used to describe malicious programs such as viruses, trojans, ransomware, bots and rootkits, are still a leading cause of cyberattacks today. This challenge has grown as the sophistication and amount of malware samples have doubled. As the number of devices has grown, the growth of the cloud and the increasing use of digital resources all have opened up a wide-open attack space. Therefore, hackers have taken advantage of advanced

obfuscation and evasion techniques, by attacking the systems that are critical at a fast rate [1]. These attackers also exploit underground markets where they exchange exploits and custom malware kits, allowing them to innovate constantly on the offensive.

Conventional detection mechanisms, especially signature-based antivirus engines, use the feature set of an analyzed file to compare against a list of existing malicious signatures. For all its decades of development, signatures have some serious drawbacks. Their fundamental flaw is their inability to detect new (zero-day) threats or very fuzzily sampled ones that are inconsistent with any known malicious patterns [2]. These new strains defy static detection with very little effort, and in many cases involve polymorphic or metamorphic changes that transform their structure but not their malicious functionality.

Similarly, as enterprise and personal computing environments are reconfigured, emerging channels of compromise — IoT devices, ICS, even mobile phones — expand the scope of attack. For most organizations, a breach can have a severe impact. In addition to direct financial losses, losses include reputational damage, regulatory fines, business interruption and stakeholder dereliction of trust [3].

In this vein, artificial intelligence (AI) has proven to be a significant companion for security analysts, filling the gap between detection power, scalability, and response speed. Machine learning (ML) models, especially deep learning (DL) architectures, make it possible to automatically extract features and dynamically classify suspicious samples. Reinforcement learning, anomaly detection algorithms, and high level hybrid algorithms further increase detection efficiency by sensing patterns that human commentators or static rules cannot detect [4].

The paper explores the history of malware analysis and how it moved from signature to AI-driven smart processes. It frames current research in the context of a wider technical and operational context, including differences between static and dynamic analysis, AI-based behavioural detection and the threat of adversaries trying to avoid or manipulate these intelligent systems. The paper also covers real-life use cases where AI-driven malware detection has been beneficial, either in the enterprise or for critical infrastructure.

Following this introduction, Section 2 is followed by a detailed literature survey on the theoretical foundations of malware detection, the impact of machine learning on multiple

paradigms of detection, and the question marks that exist around the boundaries of the field. The following sections 3–4 describe how to implement an AI-powered malware analysis pipeline, the data capture, labeling, and classification models. Part 4 presents the results of comparative evaluations of signature-based, heuristic-based and AI-based systems, and Part 5 provides the technical and practical difficulties and directions for future research. Finally, Section 6 wraps things up with some closing comments about AI's crucial contribution to strong cybersecurity.

2. Literature Review

2.1 Traditional Approaches: Signature and Heuristic Methods

Signatures were at the heart of the earliest malware analysis. Research labs like McAfee, Kaspersky, and Symantec built massive repositories of hash values and known malicious binary strings that allowed applications to detect viruses if the file it was scanning contained any similar known signature [5]. This approach works very well against known threats, but it has some major flaws. Malware writers regularly use packers, encryption, and polymorphic engines, designing variants that look distinctive from the surface, while actually being infected [6]. This means that anti-virus programs have to be updated frequently, and even a slight delay in definitions might give away newer versions.

Heuristic methods sought to overcome this deficit by analyzing suspicious behavior (e.g., specific instructions, strange API calls, or unusual file actions, i.e., self-reproducing code or exploiting existing vulnerabilities). Engines based on heuristics or intrusion detection systems (IDS) complemented signature files with rule-based reasoning, flagging out patterns. It does this by scanning for unusual system calls, looking for memory spikes, and monitoring for commands commonly present in malicious code (such as using network protocols to forward data to unrecognised IP addresses) [7].

Heuristic interpretation improved detection, but it relied on manually designed rulesets. As malware families evolved, researchers needed to constantly adjust their heuristics. This also led to false positives, notably in cases of legitimate software that executed in parallel with suspicious patterns. Furthermore, advanced attackers could "time-bomb" their malicious code

and sleep until the right moment — even without being caught in the short sandbox window of heuristic scanning [8].

2.2 Machine Learning as a Game Changer

The drawbacks of simply signature- or heuristic-based systems drove intense machine learning. Early ML-based models used statistical interpretation of program features. For example, they suggested the use of n-gram analysis on binary instruction to quantify the percentage of instruction chains associated with malicious behavior [9]. These supervised learning models, ranging from decision trees to linear classifiers, computed predictions based on features extracted from binaries or logs.

Focusing on structural and behavioural characteristics rather than isolated lines of code, ML solutions brought several benefits. They might generalise from existing examples, drawing conclusions from trends observed across thousands of malicious executables. This meant that new variants similar to known malware families were more likely to be identified. Even with these improvements, traditional ML also needed expert-based feature engineering, in which relevant features were manually selected (i.e., number of API calls, strange file types, unusual lines of code) [10].

2.3 Deep Learning and Automated Feature Extraction

Deep learning (DL) extended the ML paradigm by automating feature extraction. Neural networks, especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs), model hierarchical representations and perceive subtle correlations that manual selection would fail to notice [11]. CNNs, in particular, treat code or network data rather like image data, converting single bytes or instructions as embeddings into layer-based feature maps. RNNs, specifically Long Short-Term Memory (LSTM) networks, mapped the time history of programs' behaviour, detecting malicious patterns within chains of system calls or network packets [12].

Some studies even succeeded in using DL for static binary data. Scientists turned binaries from programs into grayscale photos, and classified them as benign or malicious through CNN-based systems [13]. Others created embeddings of mangled code and fed them into LSTM networks, and were able to pick up known and even novel malware with high

sensitivity. This transition from domain-specific heuristics to learned representations marked a significant moment in malware analysis, moving the discipline towards the flexibility and speed it needed to combat contemporary threats.

2.4 Anomaly Detection

At the same time, classification research developed anomaly detection as a strategy ideal for zero-day and advanced persistent threats (APTs). Anomaly detection systems don't classify samples as either benign or malicious; instead, they train on "normal" user, process, or network behavior. Whenever a behavior appears that falls very far from this threshold, it is flagged by the system for further analysis [14].

Isolation forests and statistical thresholding provide lightweight anomaly detection but deep learning-based models like autoencoders greatly enhance the performance by extracting the intricate dependencies of the standard data. An autoencoder method involves training a neural network to reconstruct harmless information. Anything that the network cannot reconstruct correctly is anomalous. This technique is particularly useful when looking for minor deviations, such as a valid process spawning unexpected threads or creating strange network connections.

2.5 Adversarial Attacks and Defensive Strategies

A central issue in the AI arms race is the adversarial manipulation of detection models. Cybercriminals can introduce invisible or slight changes in the code, leading classifiers to mischaracterize malicious binaries as benign [15]. Adversarial scenarios could include deliberately distorted instruction sets, or small changes in meta-information that the model views as feature values. To prevent this, scientists have tested adversarial training — training the model on samples it is deliberately manipulated to get stronger. Such techniques as defensive distillation and gradient masking, for example, likewise attempt to reduce a model's vulnerability through smoothing decision points [16].

Adversarial machine learning is a young industry, and no one has yet found the golden answer. But research still recommends a layer-based, continuously updated strategy of static and dynamic analysis, anomaly detection and robust model training as the most practical way to overcome adversarial strategies [17].

2.6 Real-World Implementations and Industrial Studies

There are numerous case studies to support the effectiveness of AI-based malware detection. Companies with networks and endpoints that span across many miles have built AI into their processes to protect high-volume payment systems. These algorithms scan millions of records and activities to discover anomalous patterns that go against the normal flow of transactions. Anomalous events trigger alerts, which leads to further research to determine whether there is malicious software or hacked credentials involved [18].

Likewise, cloud providers use AI to secure containerised services and distributed workloads. By monitoring baseline performance metrics and interactions between microservices, anomaly detection engines can identify container breaks or malicious attacks. This anticipatory approach typically mitigates breaches before they turn into massive hacks [19].

Research labs and private firms have already published countless benchmarks that show AI-based systems, when rigorously tested, produce higher detection rates for new infections at low false-positive rates. Complementary lines of evidence suggest that deep learning methods identify obfuscated or metamorphic malware more reliably than basic signature analysis [20].

3. Methodology

3.1 Analysis Framework: A Hybrid AI Pipeline

It is proposed here that we will combine static analysis, dynamic analysis, and anomaly detection into a single AI-based approach. In combining these approaches, we hope to obtain a more encompassing picture of both established and emerging risks. The mental pipeline works like this:

Data Acquisition and Labeling

It's important to have a high-quality dataset. In this experiment, we assembled a collection of more than 100,000 files, displaying nearly equal proportions of benign executables that come from legitimate sources (open source applications, operating system files) and malicious samples that range from trojans, ransomware, viruses and APT artefacts [21]. Ground-truth

labels were created by pooling responses from several antivirus vendors, reverse engineering manually, and dynamic sandbox.

Preprocessing and Feature Extraction

Static features were extracted from binary metadata (PE header information, imported and exported function lists) or partial disassembly to obtain instruction-level information before samples were fed into ML/DL models. To measure dynamic properties, each sample was run inside a simulated sandbox, which record the process's creation, changes in the registry, write operations to the file system, and network messages. The pipeline also used environment randomization to prevent sandbox detection by extremely stealthy malware [22].

Model Training

Depending on the features, different models were proposed. CNN-based models were trained on raw byte sequences and disassembly embeddings, and LSTM-based models on time series API calls. Separate modules dedicated to anomaly detection had autoencoders trained exclusively on samples that were benign, in order to detect reconstruction errors associated with malicious anomalies [23].

Validation Strategy

Cross-validation (10x) to avoid overfitting. Every fold used 90% of data for training and 10% for validation. The metrics used for performance were accuracy, detection rate, false-positive rate, and computational overhead. In addition, a second subset of 2,000 zero-day samples, extracted from newly discovered threats that weren't on hand at the time of initial data collection, was used to validate the generalisability of the trained models.

Comparison with Baseline Systems

There were three starting systems: (1) one of the most widely used commercial antivirus engines that relied on signatures and heuristics, (2) a purely heuristic engine widely used in research literature, and (3) an open-source AI engine. These baselines demonstrated how good and bad the hybrid pipeline would be in the short run.

3.2 Static and Dynamic Analysis Details

Static Analysis Tools and Techniques

In static analysis, the pipeline scanned the file headers, section entropy, and import tables for signatures of known malicious activity. Particularly high sections could suggest packing or encryption. Malware disguises itself with benign library imports, so the AI engine wasn't just using dangerous import signatures. Rather, it took into account many attributes such as control-flow graphs (CFGs) extracted from uncompressed code [24].

Dynamic Analysis Tools and Techniques

Dynamic analysis was performed in a limited sandbox which emulated a Windows 10 instance, but randomly spit out artifacts (such as CPU IDs, registry keys, timestamps) to minimize sandbox detection. The system recorded low-level actions including thread creation, injection, attempts to elevate privileges, and suspicious network requests (e.g., sending messages to known malware command-and-control servers) [25]. If the sample never woke up or identified the sandbox, sophisticated triggers (such as user interaction simulation) were used to force execution. The behavioral logs were then transformed into temporal time series data for the RNN-based classifier.

3.3 Anomaly Detection Module

In order to detect anomalies, an autoencoder was trained only on benign examples from a set of application domains (productivity apps, operating system tools, standard business apps). The autoencoder was trained to reconstruct the normal behavior of system calls and resource usage for these harmless programs. When the system found a sample whose reconstruction error exceeded a certain point, it marked the sample as anomalous and performed further evaluation [14].

This anomaly layer was particularly helpful in identifying zero-day threats and variants of malware that didn't match any of the known malicious profiles. The integration with the classification layer is part of a double-edged sword: anomalous files that don't fall under the classification logic can be flagged as abnormal, and removed for manual review.

4. Results and Discussion

4.1 Quantitative Findings

A large-scale experiment was conducted to measure the efficacy of the hybrid AI pipeline. Table 1 shows the detection performance of four systems — the best-of-the-best commercial antivirus (AV), a shared heuristic-based engine, an open-source AI engine with a minimal feature set, and the recently developed hybrid AI pipeline that includes static, dynamic and anomaly detectors.

Table 1. Comparative Performance of Malware Detection Systems

Method	**Overall Detection Rate (%)**	**Detection Rate for Zero-Day (%)**	**False Positive Rate (%)**	**Avg. Analysis Time (ms)**
Signature-Based AV	91.0	56.2	0.7	10
Heuristic-Based Engine	94.6	68.1	1.4	25
Basic AI Engine (Open-Source)	96.5	79.0	1.3	45
Proposed Hybrid AI Pipeline	**98.2**	**91.3**	**1.1**	60

Based on the collected data, we can see that the hybrid AI pipeline is more effective with a detection rate of 98.2% for the entire corpus than the other three. It has a classification accuracy of 91.3% in zero-day samples, compared to the base AI engine's 79.0% and much higher than signature-based AV's 56.2%. These observations corroborate prior work on AI's flexibility to intelligently identify malicious acts according to learned rather than pre-established rules [9, 10].

Although the proposed system's sample-analysis time (60 ms) is longer than the signature-based AV (10 ms), the trade-off seems fair for highly secure systems. The false-positive rate (1.1%) is slightly higher than the signature approach but still within a suitable range for enterprise use since the system is much more capable of detecting new threats.

4.2 Analysis of Polymorphic and Metamorphic Malware

Polymorphic and metamorphic malware presents a huge challenge to defenders. Table 2 shows the detection percentage when compared with an expert dataset of 800 malware samples that attempt to avoid static analysis through aggressive obfuscation, code morphing, and dynamic packers.

Table 2. Detection of Polymorphic and Metamorphic Malware

Method	**Detection Rate (%)**
Signature-Based AV	48.2
Heuristic-Based Engine	60.7
Basic AI Engine (Open-Source)	88.4
Proposed Hybrid AI Pipeline	**93.6**

The proposed pipeline's combination of static structural information, dynamic behavioural detection and anomaly detection significantly enhances evasion capabilities. Imitation can obscure signatures or heuristics, but dynamic analysis can still detect suspicious interactions with the host system. In the meantime, if memory usage, registry modification or lifecycle time anomalies occur, the autoencoder module reports them.

This combination accounts for the system's 93.6 % detection rate for advanced polymorphic and metamorphic attacks. Signature-based techniques, by contrast, have their detection limit reduced to near random (48.2%) because the morphing behavior of the malware bypasses static identifiers [2].

4.3 Observations on Anomaly Detection Efficacy

An anomaly detection module is one of the system's key features. We isolated more than 2,000 newly discovered zero-day samples and analyzed how many times the autoencoder reported them. The approximately 1,720 of these samples were marked as anomalous, which provided an 86% success rate before cross-checking with the classification layer. After performing further analyses, the total detection rate exceeded 90%, consistent with the integrated pipeline results reported in Table 1.

A fascinating observation was when looking at false positives. Genuine programs with weird, or recent, features set off false positives in the anomaly detection system. Other tools used for cryptography, for example, or for advanced data compression occasionally produced patterns not typical of the normal baseline. These false positives were then fed back into the classification modules to minimize misclassifications [26]. This meant that even though the anomaly detection itself occasionally increased the false-positive rate, the combined architecture helped prevent these over-alerts.

4.4 Adversarial Attack Scenarios

To assess robustness against adversarial attack, they analyzed 1,000 malicious samples modified by adversaries. Hackers had injected very little noise into code lines or metadata fields. Simple attacks such as these added random instructions that did not modify the malware's behaviour at all, but were intended to confuse the classification algorithm [15].

Observations revealed that although signature and heuristic engines were easily deceived by these sanitized samples, the new pipeline did catch around 89 % of them. The dynamic analysis was helpful because the few static tweaks made didn't change the malicious behavior at runtime. Autoencoder-based anomaly detection was another line of defence, especially for micro-manipulations that wished to stay statically hidden. But some of the most sophisticated adversarial manipulations – especially those that change dynamic behaviour to look benign – did have some effectiveness in avoiding detection. Such advanced adversarial examples make clear that AI defenses, even at a level that is substantially enhanced, are not foolproof and require continuous improvement [16].

4.5 Practical Considerations

Deployment in the real world requires an appropriate balance of computational power, detection latency, and user experience. The detection rate of the envisioned system is great, but the average analysis time of each sample (60 ms multiplied by thousands of files or streams) can add up quickly. Deployments can reduce this overhead by introducing multi-level scanning: a fast signature pass for known threats, followed by AI based analysis for suspicious or unknown files [27].

Additionally, the system's dependency on training data requires frequent updates. Malware keeps changing and an old design can get worse over time. Continual or incremental learning — coupled with threat intelligence feeds — keeps the classification and anomaly detection components in tune with new attack vectors.

5. Challenges and Future Directions

5.1 Adversarial Machine Learning and Model Robustness

Malicious actors have started building malware specifically to bypass AI by finding vulnerabilities or "blind spots" in simulated representations [15]. These attacks include gradient-based manipulations that destructively modify features that deep networks rely on. Attacks like adversarial training add distorted samples to the training data and make the model stronger [16].

However, aggressive tactics will increase as attackers develop increasingly sophisticated ways to disguise or rework malware both in place (static code) and on the move (dynamic behaviour). This cat-and-mouse scenario hints at a need for ongoing research in adversarial machine learning, encompassing effective feature extraction, robust model models, and dynamic adversarial testing techniques, which evolve in response to the emergence of new threat types.

5.2 Interpretability and Explainable AI

Deep neural networks are typically "black boxes" and security analysts never quite know the reasoning behind these categories. This opaqueness is harmful to compliance, auditing and

trust. Whether it's LIME, attention mechanisms, or saliency maps, these approaches are an attempt to explain which features generated a malicious (or benign) classification [28].

Explainability meets regulatory and legal requirements, not just for finance or medicine, but also helps human experts define detection criteria. Analysers can now check which system calls or binary offsets provoked suspicion, allowing them to foresee possible new ways of evading detection. Thus, the focus of future research should be on developing models that are accurate and decipherable.

5.3 Continuous Data Pipelines and Real-Time Detection

Modern cyberattacks demand detection and response in near real time because of their speed. In reality, a large number of enterprises use streaming analytics tools that analyze enormous amounts of logs, packets and system activity over subsecond time spans [29]. Integrating AI-based malware detection into these streaming pipelines poses challenging engineering challenges. Models should be designed with a minimum latency while keeping sufficient depth to extract all samples in-depth.

Methods such as micro-batching, approximate computation and hardware accelerators (GPU/TPU clusters) can also enable real-time deployment, but they do necessitate architecture planning and resources. In the future, we might use edge computing for initial scanning, sending only suspect information to cloud-based deep learning algorithms. These decentralized techniques help reduce network bandwidth requirements and accelerate detection by doing initial analysis closer to the datapoint [30].

5.4 Federated Learning for Collaborative Defense

Cybersecurity is intrinsically collaborative. Most enterprises can get away with pooling their threat intelligence, analyzing attacks that others have experienced, and sharing anonymous malware samples or indicators of compromise. Federated learning enables training models from multiple, distributed data sources without actually communicating raw data [31].

This method can limit data privacy and regulation issues, while maintaining better detection efficiency. All those engaged, whether hospital, bank or state, are advancing the development of a global model that's more relevant and sustainable. Federated learning is inherently

distributed, however it also demands sophisticated coordination, safe aggregation protocols, and protection against poisoning attacks from shady players [32].

5.5 Regulatory and Ethical Considerations

AI-based malware detection needs to be flexible enough to adapt to the changing regulatory landscape. Some privacy laws, like the General Data Protection Regulation (GDPR) in Europe, do constrain the manner in which data about individuals can be collected or processed, especially when private personal data is at stake. As AI systems frequently involve massive recording and analyzing of users' behaviour, it is essential to make sure these policies are adhered to [33].

The second ethical issue concerns false positives, which can prevent legitimate use of software, or essentially endanger a developer's reputation. Finding the right balance between detection and experience is an ongoing challenge, with transparent error handling and robust appeals when software is detected incorrectly flagged.

6. Extended Discussion: Real-World Case Studies and Sector-Specific Applications

AI-based malware analysis implementations are largely different from one industry to the next. Financial services, healthcare, critical infrastructure, and SMBs all have distinct risk factors and limitations. This chapter focuses on case studies and specificities, giving a more detail overview of how AI solutions are implemented.

6.1 Financial Institutions

The large number of transactions made by banks and payment processors make them ideal targets for sophisticated malware that steals credentials or steals money. AI-based solutions here tend to go beyond endpoint monitoring and include transaction monitoring. Deep neural networks reads user login patterns, transaction amounts, and geolocation data to identify anomalies that may indicate malicious activity [34].

As financial institutions have rigorous regulatory structures in place, solutions need to work with risk management tools and offer audit-friendly reports that can be inspected in the event

of fraud. Central banks and industry consortiums occasionally share threat intelligence in order to develop a holistic model of the attack landscape. Federated learning is one of the most promising methods where multiple banks can train a global threat detection model together without exposing sensitive customer information [31].

6.2 Healthcare and Medical Devices

Medical settings contain personally identifiable patient information across interconnected systems, from hospital networks to diagnostic devices and implantation devices. Ransomware attacks on this domain can disrupt critical functions and potentially even kill people. Thus, modern malware prevention is paramount [35].

Healthcare devices (mostly those in hospitals) are also a challenge as their firmware updates are scarce, and conventional antivirus solutions might not be possible due to real-time constraints and FDA certifications. AI anomaly detection can be integrated at the network level to monitor traffic patterns and detect rogue attempts to update device firmware or leak patient information. In such situations, interpretability becomes vital to follow health regulations and act responsibly in protecting patient safety [36].

6.3 Critical Infrastructure and Industrial Control Systems

ICS control everything from energy grids to assembly lines. Strikes against these systems – including Stuxnet – have demonstrated the devastating effect of malware designed to interfere with or stop industrial activities [37]. Signature-based approaches don't work because ICS systems typically implement proprietary protocols and hardware. Hackers capitalize on these quirks by using highly sophisticated malware that can overcome orthodox defences.

AIS also aids in identifying "normal" operating scenarios in ICS networks based on the baseline pressure, temperature, flow or other process parameters. A tamper detector can detect abnormalities that may suggest malicious manipulation. Also, deep learning can scan firmware images or ladder logic for anomalies that could indicate malicious code. To implement these protections, you must combine high security with the reliability requirements of a factory, where false positives disrupt the flow of production and bring enormous financial loss [38].

6.4 Small and Medium Enterprises

While large enterprises may have the capacity to deploy advanced AI-powered defences, SME's are limited in budget and talent. Small businesses are also prone to bogus ransomware or trojans that take advantage of obsolete software or configurations. In spite of these limitations, SMEs can take advantage of cloud AI services that provide robust malware detection without heavy on-premises investments [39].

In reality, lightweight agents on endpoints could forward rogue files or actions to a cloud-based analysis engine. This model utilises aggregated threat intelligence from multiple SMEs and thus acts as a kind of crowdsourced detection. The challenge is to keep the privacy of data under control, while keeping the price structure low for small companies. However, the technique democratises AI-powered cybersecurity, which could potentially increase the resilience of the larger digital community.

7. Conclusion

The ongoing battle against malware requires sophisticated, responsive, and integrated security solutions. AI-powered malware analysis remains the cornerstone of the new approach to cybersecurity, filling the inflection points that signatures and heuristics no longer reliably resolve when dealing with obfuscation, zero-day attacks and polymorphic code. Static and dynamic analysis, anomaly detection, and feature extraction via deep learning enable security systems to classify advanced malware strains in the same level of accuracy.

The empirical testing described in this paper shows the AI-based pipelines as a dominant detection mechanism, especially when it comes to new threats. By recording subtle patterns in binary configuration, execution and system-wide anomalies, these pipelines are not just static — they can keep up with evolving threats at the code and behavioral levels. Meanwhile, obstacles remain. Insidious attackers can use machine learning models to exploit vulnerabilities and spawn an endless arms race. The ever-rising importance of understandable and explainable AI is vital, both to meet the needs of regulators and to allow human analysts to work productively with AI platforms.

In the future, AI-enabled malware detection seems to revolve around model evolution, federated learning for shared threat intelligence, and application-agnostic models for healthcare, financial, and industrial environments.

References

[1] M. Alazab, S. Venkataraman, and P. Watters, "Towards understanding malware behaviour by the extraction of API calls," in *2010 Second Cybercrime and Trustworthy Computing Workshop*, IEEE, 2010, pp. 52–59.

[2] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.

[3] A. M. Al-Barashdi, S. Bouktif, and O. Zaïane, "A systematic literature review of machine learning approaches in phishing detection," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 3, pp. 3360–3369, 2020.

[4] S. Hou, K. Chang, and C. Wu, "Deep neural network-based malware detection using two-dimensional gray images," *IEEE Access*, vol. 8, pp. 56045–56059, 2020.

[5] F. Cohen, "Computer viruses: theory and experiments," *Computers & Security*, vol. 6, no. 1, pp. 22–35, 1987.

[6] T. K. Dasaklis, F. Casino, G. Patsakis, I. Chatzigiannakis, M. Piromalis, and C. Xenakis, "Defending against advanced persistent threats in a 5G world," *Electronics*, vol. 10, no. 12, p. 1492, 2021.

[7] M. Sikorski and A. Honig, *Practical Malware Analysis*. San Francisco, CA: No Starch Press, 2012.

[8] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, p. 102419, 2020.

[9] M. Schultz, E. Eskin, F. Zadok, and S. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 2001, pp. 38–49.

[10] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating shallow and deep networks for malware detection," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2018, pp. 1–6.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] S. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in Proceedings of the Australasian Conference on Artificial Intelligence, 2016, pp. 137–149.

[13] S. Nataraj, M. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, 2011, pp. 1–7.

[14] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.

[15] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[16] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy*, 2016, pp. 582–597.

[17] T. Chung, D. Choffnes, M. Sullivan, F. Li, D. Levin, B. M. Maggs, and A. Mislove, "Measuring and applying invalid SSL certificates: The silent majority," in *Proceedings of the 2016 Internet Measurement Conference*, 2016, pp. 527–541.

[18] Verizon, "Verizon 2021 Data Breach Investigations Report," Verizon Enterprise, 2021. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[19] Y. Vorontsov, E. M. G. Rodrigues, K. Kim, and W. Lin, "Using machine learning approaches for cloud intrusion detection," *IEEE Access*, vol. 9, pp. 157698–157710, 2021.

[20] E. Stinson and J. C. Mitchell, "Characterizing bots' remote control behavior," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 4579, LNCS, 2007, pp. 89–108.

[21] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123–147, 2019.

[22] M. Egele, D. Kirda, C. Kruegel, and G. Vigna, "Dynamic malware analysis in the modern era—A state of the art survey," *Journal of Information Security and Applications*, vol. 79, no. 3, pp. 186–210, 2012.

[23] L. Meng, T. Jiang, and R. H. Deng, "When intrusion detection meets deep learning: A review," *IEEE Access*, vol. 8, pp. 106180–106202, 2020.

[24] A. Rajabzadeh, P. Franke, and M. Conti, "Binary function similarity detection in software engineering and malware analysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4059–4073, 2021.

[25] F. Audi, J. E. Tapiador, and P. Peris-Lopez, "Trends and challenges in anomaly-based intrusion detection of IoT traffic: A comprehensive survey," Sensors, vol. 20, no. 18, p. 5227, 2020.

[26] X. Xiao, Y. Zhang, J. Wu, and H. Feng, "An efficient feature selection method for malicious traffic detection in IoT networks," *Future Generation Computer Systems*, vol. 114, pp. 375–388, 2021.

[27] J. Wang, Q. Chen, and Y. Yang, "Adaptive security solutions in cyber-physical systems through multi-level scanning," *Computers & Security*, vol. 105, p. 102236, 2021.

[28] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[29] C. S. Collberg and S. Kobourov, "Streaming analytics for big data cybersecurity: A survey," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–40, 2021.

[30] Y. Zhao, J. Li, and F. Xu, "Edge computing and security in the era of IoT: A systematic review," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10409–10430, 2021.

[31] K. Yang, T. Jiang, and Y. Shi, "Federated machine learning for intelligent IoT via reconfigurable intelligent surfaces," IEEE Network, vol. 35, no. 5, pp. 16–22, 2021.

[32] A. Bhagoji, D. Cullina, and P. Mittal, "Dimensionality reduction as a defense against poisoning attacks on machine learning classifiers," in Proceedings of the 29th USENIX Security Symposium, 2017, pp. 343–360.

[33] A. Chouldechova and A. Roth, "The frontiers of fairness in machine learning," *Communications of the ACM*, vol. 63, no. 5, pp. 82–89, 2020.

[34] B. L. Mirkin, "AI in financial transactions: Fraud detection and compliance," *IBM Journal of Research and Development*, vol. 65, no. 4/5, pp. 1–10, 2021.

[35] A. Kelarestaghi, K. Salah, and M. Conti, "Ransomware propagation in healthcare networks: Detection using AI-based solutions," *IEEE Network*, vol. 35, no. 4, pp. 123–129, 2021.

[36] S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[37] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," *Symantec Security Response*, vol. 5, pp. 1–69, 2011.

[38] T. T. Oh, M. A. Ngadi, I. Ahmad, J. E. Abawajy, and C. Su, "Anomaly detection and classification in modern ICS using deep belief networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3432–3442, 2021.

[39] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.