electronics

MDPI

Article

Machine Learning for Cybersecurity: A Survey of Applications, Adversarial Challenges, and Future Research Directions

Zefeng He, Diego Davila, Shengping Bi, Tao Wang, and Tao Hou

Department of Computer Science and Engineering, University of North Texas;
{zefenghe@my., diegodavila@my., shengpingbi@my., tao.wang2@, tao.hou@}unt.edu

Abstract

The convergence of ubiquitous connectivity, large-scale data generation, and rapid advancements in machine learning is transforming the field of cybersecurity. The widespread adoption of interconnected systems including Internet of Things devices, mobile platforms, and cloud infrastructures has introduced new attack surfaces and significantly increased the complexity of securing digital environments. Concurrently, these technologies have enabled the development of intelligent, data-driven defense strategies. Achieving effective protection in these settings requires not only applying machine learning to detect and prevent threats but also recognizing that such models can themselves become targets of adversarial manipulation. This survey presents a comprehensive analysis of recent progress at the intersection of machine learning and cybersecurity. It explores defensive applications such as malware detection, network traffic classification, and anomaly detection, as well as offensive strategies including adversarial evasion, poisoning, and backdoor attacks. Particular attention is paid to adversarial machine learning, highlighting the increasing sophistication of attacks that exploit model vulnerabilities and the corresponding evolution of defense mechanisms. Beyond synthesizing current research, the survey also identifies key open challenges and emerging research directions. This survey provides a comprehensive and accessible reference for researchers and practitioners aiming to understand and advance the secure application of machine learning across diverse cybersecurity domains.

Keywords: Adversarial Machine Learning; Trustworthy Machine Learning; Responsible AI; Malware Detection; Network Traffic Analysis; Anomaly Detection.

Received:
Revised:
Accepted:
Published:

Citation: . Machine Learning for Cybersecurity: A Survey of Applications, Adversarial Challenges, and Future Research Directions. *Electronics* **2025**, *1*, 0. https://doi.org/

1. Introduction

Machine learning refers to a class of computational methods designed to build models that learn from data in order to perform specific tasks, without relying on manually defined rules or heuristics. By optimizing performance over large datasets, these models are capable of identifying intricate patterns and relationships, thereby supporting both predictive and descriptive capabilities. Consequently, machine learning has achieved notable success across diverse fields, including natural language processing, computer vision, financial modeling, and autonomous systems [20,26,27]. With the continued expansion of data availability and computational power, machine learning has become a cornerstone in both academic research and industrial practice.

At the same time, cybersecurity has undergone a profound transformation. Modern information systems are now highly interconnected, heterogeneous, and increasingly defined by software, driven by the widespread adoption of smart devices, cloud computing platforms, and edge-based architectures. This evolution has dramatically expanded the

attack surface, exposing digital infrastructure to more sophisticated, stealthy, and large scale threats [29]. Traditional security mechanisms, such as rule-based firewalls, manually configured intrusion detection systems, and signature-based malware scanners, are proving insufficient in addressing these emerging challenges. Their reliance on static rules and known threat patterns limits their ability to detect novel, evasive, or zero-day attacks [28]. As a result, there is an urgent need for intelligent, adaptive, and data-driven security approaches capable of learning from dynamic environments and responding proactively to evolving threats.

Machine learning offers powerful tools for addressing emerging cybersecurity challenges. By enabling systems to learn from past behaviors, process large volumes of data, and make timely decisions, machine learning techniques have demonstrated strong potential across a range of security applications. These include intrusion detection, malware classification, anomaly detection, network traffic analysis, vulnerability discovery, and behavioral modeling. For instance, machine learning-based intrusion detection systems can distinguish between benign and malicious traffic by identifying patterns in network flows. Similarly, malware classifiers can detect harmful executables by analyzing either static code features or dynamic runtime behaviors [30,31]. Unlike traditional rule-based methods, machine learning models are capable of adapting to new threats and often maintain high accuracy across varied datasets and operational environments.

The growing adoption of machine learning in cybersecurity is further fueled by the exponential increase in data volume and complexity. Modern digital systems, including enterprise servers, mobile devices, and embedded IoT platforms, continuously generate large streams of high-dimensional and often unlabeled data. Extracting meaningful insights from this data in real time requires scalable, noise-tolerant, and adaptive algorithms capable of handling non-stationary distributions with low latency. Deep learning models, such as convolutional neural networks, recurrent architectures, and transformers, have demonstrated strong performance in these settings. Recent studies [20,27] have also applied transfer learning and pretraining techniques, originally developed for natural language processing, to binary code analysis and malware detection with promising results.

However, the convergence of machine learning and cybersecurity also introduces significant risks. Despite their capabilities, machine learning models are inherently vulnerable to various forms of adversarial manipulation [32]. Attackers can craft adversarial examples that subtly perturb inputs to mislead classifiers, poison training data to degrade model performance, or extract sensitive information from trained models through inference attacks [61]. In some cases, entire models can be compromised by inserting backdoors or trojans. These vulnerabilities create novel attack vectors that target the learning algorithm itself rather than the systems it is designed to protect. Moreover, adversaries are increasingly using machine learning to amplify the effectiveness, stealth, and scalability of their attacks, creating a dual-use dynamic in which both attackers and defenders rely on similar technologies.

The rapid evolution of cyber threats and the growing arms race between defensive and offensive uses of machine learning highlight the urgent need for a comprehensive understanding of how these technologies are applied, evaluated, and secured. As attackers develop more sophisticated and adaptive methods, defenders must continually evolve their tools and strategies to keep pace. Machine learning has become both a powerful enabler of next-generation defense systems and a new target of attack. This dual role introduces complex challenges, including ensuring the robustness, interpretability, and trustworthiness of ML-driven security mechanisms. For students, researchers, and practitioners in cybersecurity, it is no longer sufficient to simply apply machine learning algorithms; they

must also anticipate adversarial behaviors, identify hidden failure modes, and design systems that can operate securely in dynamic, high-stakes environments.

This survey aims to provide an accessible, technically grounded synthesis of the current state of research at the intersection of machine learning and cybersecurity. It is intended to serve both as a resource for those new to the field and as a reference for experienced researchers seeking to explore new directions. We present a structured overview that spans fundamental concepts, key applications, and emerging challenges, with a focus on practical relevance and real-world impact. By integrating insights from recent academic publications and experimental studies, the survey identifies patterns, exposes gaps, and highlights future research opportunities. In doing so, we hope to foster a deeper understanding of how machine learning can be leveraged to improve cybersecurity outcomes while remaining aware of the risks it introduces.

To structure the discussion, this survey is organized around three core domains that represent key areas where machine learning intersects with cybersecurity:

- **Malware Detection:** We examine how machine learning techniques are used to identify malicious software by analyzing both static features, such as byte sequences and API call graphs, and dynamic behaviors, including runtime system interactions. This survey covers a range of models, from traditional classifiers to modern deep learning architectures and transformer-based approaches.
- **Network Traffic Analysis and Anomaly Detection:** We review machine learning approaches for identifying suspicious or abnormal activity in communication networks. These methods are especially important for securing distributed environments and IoT ecosystems, where conventional intrusion detection systems often fall short. Techniques discussed include unsupervised anomaly detection, deep autoencoders, clustering methods, and federated learning for decentralized threat detection.
- **Adversarial Machine Learning:** We examine how adversaries exploit vulnerabilities in machine learning models and how defenders design countermeasures to protect them. This section covers common attack strategies, including evasion, poisoning, and backdoor insertion, as well as defensive techniques such as adversarial training, robust optimization, and input validation at inference time.

Each domain presents distinct challenges related to data representation, algorithmic complexity, interpretability, and robustness against adversarial threats. Throughout this survey, we highlight notable contributions, summarize experimental findings, and discuss current limitations, while identifying promising directions for future research. Our objective is to support the advancement of intelligent cybersecurity systems that are not only accurate but also transparent and resilient in real-world deployments.

2. Related Work

The intersection of machine learning and cybersecurity has attracted increasing research attention, resulting in a growing body of survey literature. However, most prior works adopt a narrow scope, focusing either on a specific threat model, a particular security domain, or a subset of machine learning techniques. In contrast, this survey aims to provide a comprehensive and unified perspective that spans offensive and defensive machine learning, real-world cybersecurity applications, and emerging trustworthy AI paradigms.

Early works in this space typically emphasized individual threat vectors or specific cybersecurity use cases. For example, Demetrio et al. [16] conducted a technically detailed and systematic review focusing exclusively on adversarial evasion attacks targeting Windows malware detectors. Their survey presents pseudocode-level algorithms and curated benchmarks, offering valuable insights into low-level adversarial manipulation techniques. Nevertheless, its emphasis on a single OS ecosystem and evasion-only threat model limits

its applicability to broader machine learning security settings, such as encrypted traffic analysis, cloud security, and secure model deployment.

Similarly, Ullah et al. [62] examined computer-vision-inspired malware analysis pipelines, where binaries are converted to grayscale images and classified using convolutional neural networks. They presented taxonomies of visual features, CNN architectures, and datasets. This perspective highlights a creative crossover between computer vision and security, but does not explore recent advances such as transformer-based binary representation learning, self-supervised malware embeddings, or adversarial training for code and binary classifiers. Furthermore, their focus is primarily on malware, without extending to modern IoT and network detection tasks.

Recent survey efforts have broadened the analysis of adversarial machine learning. Zarras et al. [16] provided a detailed overview of adversarial defenses across multiple algorithmic categories. Their survey systematically analyzes threat models, robustness strategies, and evaluation techniques, offering a structured defense taxonomy. However, the work primarily focuses on adversarial robustness and does not cover broader ML-for-security applications such as federated IoT threat detection, self-supervised traffic classification, or industrial cybersecurity systems.

Similarly, Pelekis et al. [17] examined adversarial machine learning from a cross-industry angle, emphasizing attack vectors affecting critical sectors including healthcare, transportation, and finance. Their review provides a useful industry lens and policy context, yet emphasizes the attacker-defender interplay rather than end-to-end ML system design, real-time detection challenges, and the role of large-scale representation learning in security.

Beyond academic surveys, the community has also seen standardization efforts. Vassilev et al. [18] published a NIST taxonomy that formalizes adversarial machine learning terminology, establishing a common lexicon for attacks and mitigations. This work provides valuable alignment for threat modeling and evaluation frameworks, yet it does not review empirical ML-based cybersecurity systems, emerging architectures like transformers and graph neural networks, or the privacy-utility trade-offs in federated security learning.

A complementary body of work studies narrower paradigms. Bai et al. [63], for instance, focus exclusively on membership inference attacks in federated learning, offering detailed insights into privacy leakage in distributed training. While critical for privacy-preserving security, such narrow focus does not address broader ML-enabled detection challenges or adversarial resilience in operational networks. Other task-specific surveys have explored backdoor attacks, poisoning strategies, and secure model extraction defenses, but often in isolation and primarily within computer vision tasks.

In contrast, this survey provides a unified and cross-domain synthesis. We examine (i) offensive machine learning techniques such as adversarial evasion, poisoning, model extraction, and backdoor attacks, and (ii) defensive applications across malware detection, encrypted traffic analysis, IoT anomaly detection, and cloud-scale network modeling. We further highlight emerging paradigms including transformer architectures, graph neural networks, federated and self-supervised learning, and the growing role of large language models in cybersecurity workflows. Importantly, we also discuss practical dimensions such as interpretability, scalability, data governance, and real-world deployment challenges, factors often overlooked in narrower works.

In summary, while prior surveys have contributed valuable knowledge within specific niches, no existing review provides an integrated treatment that connects modern machine learning advances, adversarial threat landscapes, and practical security deployment. By bridging these research threads, our survey aims to offer a comprehensive reference that supports both foundational understanding and the design of trustworthy, resilient, and adaptive ML-driven cybersecurity systems.

3. Review Methodology

To ensure rigor and reproducibility, we adopted a structured process for identifying, screening, and selecting the literature included in this survey. We queried major scholarly databases including IEEE Xplore, ACM Digital Library, SpringerLink, Scopus, and arXiv. The search period primarily covered the most recent ten years, capturing the rapid evolution of machine learning and adversarial machine learning in cybersecurity. Keyword combinations were formed using terms such as machine learning, cybersecurity, adversarial learning, malware detection, intrusion detection. Search expressions were adapted to each digital library, and backward and forward citation tracing was used to identify additional influential works.

Publications were first screened by title and abstract, followed by full-text review for relevance and technical depth. We prioritized peer-reviewed journal and conference papers, but also included high-impact preprints when they provided timely coverage of emerging research directions or when peer-reviewed alternatives were not yet available. When a paper appeared in both preprint and peer-reviewed form, the peer-reviewed version was retained. Studies were included if they directly addressed machine learning for cybersecurity or security for machine learning, offered a technical contribution such as a model, dataset, taxonomy, or empirical study, and demonstrated clear relevance to adversarial resilience or ML-enabled defense strategies.

Works were excluded if they lacked technical content, focused solely on traditional non-ML security techniques, repeated content from previously selected papers, or constituted commentary without methodological contribution. After screening, the remaining papers were synthesized across thematic categories aligned with this survey.

Table 1. Summary of the surveyed malware detection papers

Paper	Robustness	Architecture	Dataset	Notes
[36]	Minimal/none	Transformer	Original dataset	Embedding generation with transformers
[35]	Minimal/none	Transformer	BinaryCorp(original work)	Applicable to vulnerability detection
[48]	Minimal/none	Decision tree	Customized NPM dataset	Applied to identify unfound malicious npm packages
[40]	Minimal/none	CNN	Kaggle malware classification competition	Use of Perceptual Hashing for quick initial classification
[41]	Minimal/none	CNN/Several	CICMaldroid2020/Drebin	Comparative study of several models
[42]	Minimal/none	CNN	Malimg, Microsoft BIG 2015, Malevis	Application of pre-trained image models
[38]	Minimal/none	Transformer	Androzoo	Detected wild malware with 59.3% accuracy
[39]	Minimal/none	CNN	Filtered Large PE Malware from VirusTotal	High few-shot accuracy
[44]	Significant	GNN	CICMaldroid2020/Drebin	Developed retraining process for adversarial robustness
[45]	Minimal/none	NMF/Clustering	EMBER-2018	100% true rejection on unseen malware families.
[43]	Significant	Decision Tree	HPC dataset derived from VirusTotal	Adversarial effetiveness reduced 50-fold with retraining

4. Malware Detection

Malware refers to software intentionally designed to disrupt, damage, or gain unauthorized access to computer systems, often serving the objectives of malicious actors. Its origins

precede the modern internet, with the first known networked worm, Creeper, demonstrated by Bob Thomas in 1971 [34]. As modern computing environments become increasingly diverse, ranging from mobile devices and embedded IoT platforms to large scale cloud infrastructures, malware has evolved to become more complex, evasive, and polymorphic. This evolution presents formidable challenges to traditional detection strategies that rely on signature matching, checksum verification, or handcrafted heuristic rules. These conventional methods, while effective against known threats, require prior knowledge and often fail to detect zero-day attacks or novel malware variants.

To address these limitations, researchers have increasingly turned to machine learning to improve malware detection. These data-driven techniques learn behavioral and structural patterns from existing samples and generalize to previously unseen threats. Recent research has applied techniques originally developed in fields such as natural language processing and computer vision. NLP inspired approaches, especially those leveraging transformer models, treat binary code as token sequences, enabling the extraction of semantic representations useful for classification and similarity analysis. In contrast, vision based methods transform executable binaries into grayscale images and apply convolutional neural networks to identify spatial patterns that separate malicious files from benign ones.

Although these approaches have shown strong performance on various datasets, key challenges remain. These include detecting novel malware families with limited training examples, improving resilience to adversarial inputs, and enhancing model interpretability and operational scalability. In practice, only a few of the surveyed studies explore these robustness issues in depth.

To provide an organized comparison of existing studies, Table 1 summarizes a selection of recent machine learning based malware detection research. It outlines the core architecture used in each study, the dataset employed, the degree of adversarial robustness considered, and notable implementation notes. This structured summary helps highlight emerging trends, research gaps, and potential opportunities for future development in malware detection using machine learning.

4.1. NLP-Inspired and Transformer-Based Malware Detection

A growing trend in cybersecurity research is the adaptation of natural language processing techniques to malware detection. This approach stems from the conceptual similarity between binary code and natural language: both can be modeled as structured sequences governed by syntax and semantics. By treating code as a form of language, NLP-inspired methods can be employed to learn contextual embeddings that capture intricate program behavior, thereby improving malware classification accuracy and resilience.

Among NLP-based techniques, the transformer architecture has emerged as particularly impactful, with BERT (Bidirectional Encoder Representations from Transformers) being a prominent example. BERT is trained through two self-supervised pretraining tasks: masked language modeling (predicting masked tokens in a sentence) and next sentence prediction (determining whether two sentences appear in sequence) [26]. Its success in capturing deep contextual semantics in text has inspired a wave of adaptations for binary analysis and malware detection.

Wang et al. developed jTrans, a BERT-style model specifically tailored for binary code similarity detection (BCSD) [35]. They replaced the next-sentence prediction task in BERT with a novel control-flow jump prediction task, allowing the model to learn execution flow dependencies within binary code. This pretraining strategy yielded a 99.5% accuracy in the jump prediction task, and significantly improved BCSD performance by 7.5% over a standard BERT baseline. This work highlights how carefully selected pretraining tasks can

infuse domain-specific knowledge into transformer models, enhancing their downstream effectiveness.

Building on similar principles, Li et al. introduced PalmTree, which applies the BERT architecture to generate instruction-level embeddings from disassembled code [36]. These embeddings are designed to serve as input to downstream classifiers for malware and vulnerability detection. Experimental results showed that PalmTree embeddings outperformed prior embedding schemes across multiple benchmarks, demonstrating the utility of language modeling for capturing semantic representations of machine instructions.

Beyond binary similarity and instruction embedding, transformers have also been directly applied to malware classification. Long et al. trained a transformer-based model on Android malware samples from the Androzoo dataset, achieving a 59.3% detection rate on real-world, wild malware instances [38]. Notably, this performance was comparable to that of leading commercial antivirus solutions at the time, suggesting the feasibility of transformer-based approaches in operational environments. Unlike traditional signature-based detection, transformer models can generalize to previously unseen malware variants, making them well-suited for evolving threat landscapes.

The success of these approaches reflects a broader convergence between cybersecurity and advances in NLP. In particular, the context-aware representations produced by transformers are capable of capturing nuanced patterns in code behavior, control flow, and instruction semantics. These representations are less brittle than manually crafted features and offer better generalization across diverse malware families and evolving attack vectors.

Despite promising results, several open challenges remain. Many current transformer-based models assume access to large, labeled corpora for pretraining or fine-tuning, which may not always be available in malware domains. Moreover, limited attention has been given to evaluating the adversarial robustness of these models. For example, jTrans and PalmTree report impressive classification accuracy but do not analyze the models' resilience to adversarial examples or evasion attacks. This omission presents a potential vulnerability, as adversaries could craft subtly perturbed binaries that manipulate token representations and bypass detection.

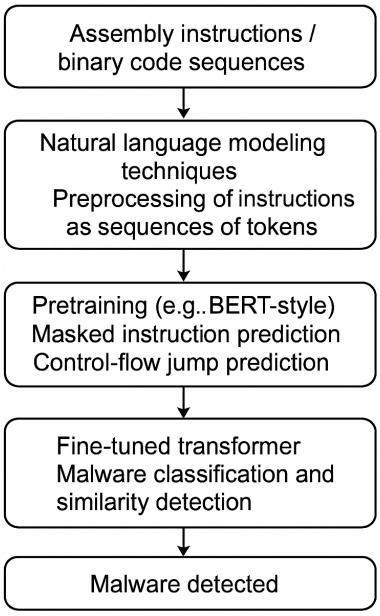

Figure 1. Workflow of NLP-inspired malware detection using transformer-based models. Raw binaries are tokenized and processed similarly to text data, enabling pre-training with NLP objectives like masked prediction and instruction sequence modeling. This allows fine-tuning for downstream malware detection tasks.

In summary, the integration of NLP and transformer architectures into malware detection represents a powerful and promising direction for both research and deployment. As illustrated in Figure 1, the typical workflow begins by treating raw binaries as sequences, analogous to natural language tokens. These sequences undergo tokenization and embedding, followed by pretraining using NLP-inspired objectives such as masked token prediction or control flow modeling. The pretrained models, typically based on transformer architectures like BERT, can then be fine-tuned for downstream malware classification tasks. This pipeline enables the extraction of rich, contextual representations from binary code, offering improved detection accuracy and better generalization across diverse malware families. As transformer-based NLP continues to advance, with developments such as RoBERTa, GPT-style models, and domain-specific pretraining objectives, future research should explore their potential to enhance robustness against adversarial samples and adaptability to emerging threats in cybersecurity.

4.2. Vision-Inspired and CNN-Based Malware Detection

Convolutional neural networks, originally developed for image classification and object recognition tasks in computer vision, have been successfully adapted for malware detection by treating binary code as image-like data. This approach leverages the spatial pattern recognition capabilities of CNNs to identify malicious behavior patterns embedded within binary files. In this line of work, malware binaries are typically converted into grayscale images where each byte is mapped to a pixel value. These representations enable deep learning models to process binary code as if it were image data, allowing for automated feature extraction and robust classification.

A notable example is DPNSA, presented by Chai et al., which introduces a dynamic convolutional architecture to address the few-shot classification challenge in malware detection [39]. Few-shot learning aims to classify samples from classes with limited labeled examples, a critical need in cybersecurity where novel malware variants often appear before large training datasets can be collected. DPNSA modifies standard CNNs to incorporate dynamic parameter adjustments, enhancing the model's adaptability to limited-data scenarios. The model achieved strong results on both 5-shot and 10-shot classification tasks, with accuracies of 88.60% and 90.28% respectively, outperforming conventional baselines by margins of 5.25% and 3.65%. This work demonstrates the practical viability of deploying deep learning models for real-world malware detection where data scarcity is a persistent issue.

Beyond standard CNNs, other approaches have incorporated hybrid techniques to improve accuracy and generalizability. Li et al. proposed PH-CNN, a model that combines perceptual hashing and CNNs for malware classification [40]. Perceptual hashing allows for the generation of hash values that are resilient to minor alterations in the input image while preserving semantic similarity. In PH-CNN, the binary image is first compared to a database of known malware families using Hamming distance on perceptual hashes. If no match is found, the CNN component is used to classify the image. This two-stage hybrid pipeline significantly enhances robustness and accuracy, achieving 98.98% accuracy on the Microsoft Malware Classification Challenge dataset. The combination of hashing and deep learning provides a compelling direction for defending against obfuscated or slightly modified malware samples.

Nguyen et al. explored additional CNN-based architectures tailored for Android malware classification [41]. Their study compared a range of models, including random forests and one-dimensional CNNs (1D-CNNs), across multiple datasets such as Drebin and CICMalDroid2020. The 1D-CNN model excelled in handling sequential byte-level data and outperformed traditional machine learning approaches on several benchmarks.

Their results highlight that CNNs can be effective not only in image-based representations of binaries but also in directly learning from sequential opcode-level or byte-level inputs, especially in resource-constrained mobile and IoT environments.

Another interesting direction involves the application of transfer learning and pre-trained vision models to malware detection. Aslan et al. proposed leveraging popular CNN architectures such as AlexNet and ResNet-50, originally trained on ImageNet, as feature extractors for binary image classification [42]. Their findings reveal that high-level visual features extracted from these models can generalize surprisingly well to malware classification tasks, especially when combined with fine-tuning techniques or additional fully connected layers. This approach benefits from reduced training time and improved performance when labeled malware datasets are limited.

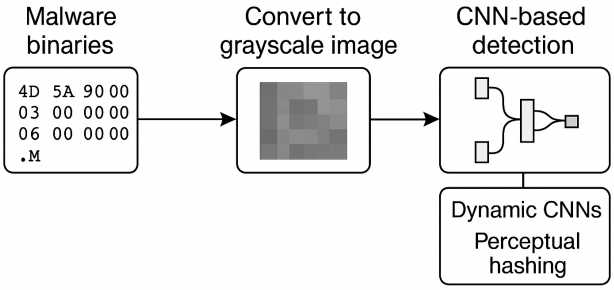

Figure 2. Workflow of CNN-based malware detection. Malware binaries are preprocessed into visual or sequential representations, which are then passed through convolutional layers to extract hierarchical features. Advanced strategies such as perceptual hashing, few-shot learning, and transfer learning enhance the system's ability to detect both known and previously unseen malware variants.

In summary, convolutional neural networks (CNNs) provide a powerful and flexible approach to malware detection by enabling automated feature extraction from both visual encodings and sequential representations of binary code. Their ability to generalize from raw byte sequences or image-like inputs allows them to detect subtle structural patterns and behavioral signatures that may be missed by traditional techniques. As illustrated in Figure 2, the typical workflow involves converting malware binaries into grayscale images or byte sequences, extracting hierarchical features through convolutional layers, and performing classification using fully connected layers or specialized decision modules. Enhancements such as perceptual hashing, few-shot learning frameworks, and transfer learning further boost the model's ability to detect obfuscated or previously unseen malware families. Continued innovation in CNN-based architectures, especially in combining them with hybrid models or hierarchical learning strategies, holds strong promise for tackling the evolving and adversarial nature of malware threats in practical deployments.

4.3. Challenges

Despite impressive advances in machine learning-based malware detection, several challenges continue to limit the effectiveness and real-world deployment of these models. Two key issues stand out: the difficulty of detecting novel or low-shot malware families, and the limited attention given to adversarial robustness.

One major limitation in current approaches is the reliance on many-shot classification tasks. Most surveyed works utilize datasets where each malware family is well-represented. For example, in the Drebin dataset used by Nguyen et al. [41], the smallest class, "Geinimi," still contains 92 samples, while "FakeInstaller" has 925 samples. However, in real-world scenarios, newly emerging malware families may initially be represented by only a handful of samples, or none at all. This makes few-shot or zero-shot generalization a critical capability for practical detection systems. Only a few papers, such as Chai et al. [39], attempt to

tackle this through dynamic convolutional architectures for few-shot learning. Yet even these approaches may struggle with entirely novel malware generated by polymorphic or metamorphic engines.

Generalization challenges are not limited to data scarcity. The ability to transfer learned features across different malware types or platforms is still underdeveloped. Wang et al. [35] demonstrate that pretraining on binary code control flow tasks (e.g., jump prediction) improves similarity detection, suggesting that transferable learning objectives can help. Likewise, Aslan et al. [42] leverage pretrained vision models (AlexNet and ResNet-50) for feature extraction on malware binaries. In another direction, Eren et al. [45] apply non-negative matrix factorization (NMF) to extract latent feature matrices for clustering, achieving 100% true rejection rate for novel malware. Note that the result reported under controlled hold-out evaluation where novel malware families were excluded during training, and performance may differ in real-world deployment due to malware diversity, evolution over time, and dataset representativeness. These results indicate that there exist learnable, reusable representations in the binary domain, but the research community still lacks a standardized approach to extract and utilize them effectively.

A second, equally pressing challenge is the lack of robustness against adversarial attacks. As shown in Figure 3, most surveyed studies provide little or no analysis of how their models perform under adversarial perturbations. This oversight is particularly concerning given that attackers can subtly manipulate binaries to evade detection without compromising malware functionality.

Among the few exceptions, Elnaggar et al. [43] explore adversarial robustness using a tree-based detection model trained on hardware performance counters. Their approach incorporates a robust score function during tree construction, accounting for all possible perturbations at each node split. This method improves resilience while maintaining interpretability and scalability, though it is limited to tabular data modalities.

Yumlembam et al. [44] adopt a different strategy by retraining Graph Neural Networks (GNNs) with adversarial examples injected into the training set. While this improves robustness, it slightly degrades clean-data accuracy. Such trade-offs remain a common challenge in adversarial defense, particularly in malware detection where minor perturbations may not be semantically meaningful or reversible.

Figure 3 provides a visual overview of the extent to which different papers incorporate adversarial robustness. It highlights a critical research gap and calls for the integration of defense-aware evaluation metrics into malware detection benchmarks.

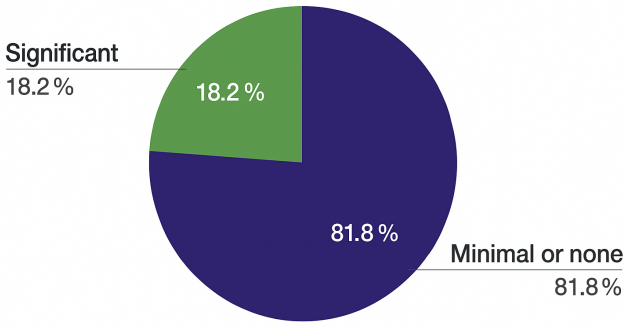

Figure 3. Surveyed malware detection models and their incorporation of adversarial robustness. Most lack defense mechanisms or evaluation against adversarial examples.

In conclusion, while current machine learning-based malware detection methods achieve strong performance on benchmark datasets, they must evolve to address real-world threats. Future work should emphasize low-shot generalization, robust feature learning, and adversarial defenses. Exploring combinations of transformers, CNNs, and GNNs with

Table 2. Summary of the surveyed network traffic analysis and anomaly detection papers

Name	Model	Dataset	Type	Notes
[37]	Transformer or BERT	CSTNET	Encrypted traffic classification	Modified natural language pretraining model BERT
[51]	Dense NN	UNSW-NB15	Federated anomaly detection	Privacy protection for potential IoT application
[47]	DFT, SMT	Several	Realtime anomaly detection	Operates in the frequency domain instead of packet-level
[46]	Tree and Forest	Original	Encrypted traffic classification	Able to identify DNS over HTTPS and browser type
[28]	Autoencoder	Original	Anomaly detection for IoT botnets	Produced dataset from actual botnet infected network of devices
[49]	Decision Trees	Several	Model interpretation approach	Performs interpretable model extraction

robust optimization and self-supervised learning may offer promising directions to bridge these gaps. .

5. Network Traffic Analysis and Anomaly Detection

As encrypted communication protocols such as HTTPS, TLS, and DNS over HTTPS become increasingly widespread, traditional rule-based traffic inspection tools face mounting limitations in understanding and classifying network activities. Since the payloads of such traffic are hidden, conventional signature or pattern matching methods often fall short. To address this gap, researchers have explored machine learning techniques capable of extracting meaningful patterns from encrypted or anonymized traffic to support security analysis and threat detection.

The machine learning approaches surveyed in this section broadly fall into two overlapping categories. The first is network traffic analysis, which focuses on identifying and classifying traffic patterns based on observable characteristics like flow timing, burst structure, and packet size distribution, even when payload contents are inaccessible. For example, encrypted bursts of traffic can be analyzed to infer the originating application or detect covert channels [37,46].

The second category is anomaly detection, which seeks to learn a representation of normal or benign network behavior and then flag deviations that may signal malicious activity, misconfigurations, or previously unseen threats. These models typically rely on unsupervised or semi-supervised techniques such as auto-encoders or clustering based on frequency domain features to detect irregularities in real time [28,47].

Both types of techniques demonstrate that even in the absence of decrypted payloads, statistical signals and structural metadata in network traffic can be leveraged to draw security relevant conclusions. This capability is becoming increasingly essential in modern cybersecurity environments, where privacy preserving protocols and sophisticated adversaries coexist.

To provide an overview of the contributions discussed in this section, Table 2 summarizes the key characteristics of the reviewed papers, including the models used, datasets employed, analysis type, and notable technical features. The subsections that follow offer a more detailed discussion of recent advances in traffic analysis and anomaly detection, followed by a discussion of cross-cutting challenges related to interpretability, scalability, and adversarial resilience.

5.1. Network Traffic Analysis

With the proliferation of encryption protocols, conventional deep packet inspection techniques are increasingly rendered ineffective. While encryption preserves confidentiality, it also complicates the task of traffic classification and threat detection for cybersecurity practitioners. Consequently, machine learning models have emerged as powerful tools to infer meaningful patterns and application level semantics from encrypted traffic without decrypting the payload. This subsection reviews recent advances in using ML for encrypted traffic analysis, demonstrating that even limited observable features such as packet size, timing, and sequence can provide rich input for inference.

Transformer Based Traffic Representation. Lin et al. [37] propose a novel application of the BERT transformer architecture, originally developed for natural language processing, to the classification of encrypted network traffic. Rather than textual tokens, their model ingests sequences of low-level packet features such as size, direction, and interarrival timing. The pretraining process is inspired by masked language modeling, where a portion of the input sequence is masked and predicted by the model. Notably, they replace 10 percent of the masked tokens with randomly selected alternatives rather than a static mask token. This design choice encourages the model to better generalize in the presence of real-world noise.

In addition to standard masked prediction, Lin et al. introduce a self supervised task called same origin detection, wherein the model must infer whether two segments or bursts of traffic originate from the same application source. Importantly, all pretraining is conducted on entirely unlabeled traffic data, which showcases the feasibility of self supervised learning in domains where labeled examples are scarce or unavailable.

Figure 4 provides an overview of this architecture. Packet-level features serve as input to the transformer model, which undergoes self supervised training before fine tuning for inference tasks such as application classification or randomness analysis. The resulting model not only achieves superior performance compared to ten state of the art baselines in both few-shot and many-shot scenarios, but also demonstrates strong generalization capabilities. This includes promising performance on auxiliary tasks such as entropy estimation, reinforcing the utility of pretrained transformer based models for encrypted traffic representation.

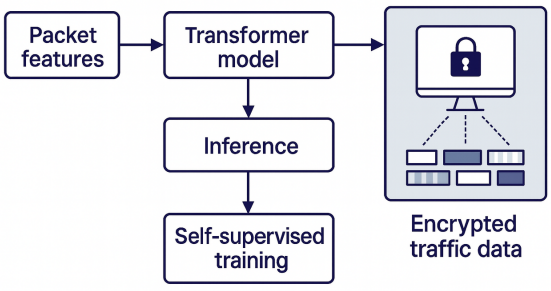

Figure 4. Transformer based architecture for encrypted traffic representation. The model is pretrained on sequences of packet features using self supervised tasks, followed by fine tuning for downstream inference.

ML Based Classification of DoH Traffic. In another work, Vekshin et al. [46] examine the classification of DNS over HTTPS traffic, a protocol often exploited for covert communication. They evaluate five widely used ML classifiers, including AdaBoost and Random Forest, to determine their efficacy in classifying DoH traffic. Notably, AdaBoost and Random Forest achieved near perfect performance, both reporting 99.9% accuracy.

Their model is not only able to detect traffic anomalies but also classify the client application (e.g., Chrome, Cloudflared, or Firefox) generating the traffic, achieving recall scores

of 99.0%, 99.9%, and 98.9%, respectively. These results highlight that despite encryption, significant application level fingerprints persist in traffic metadata. However, the authors acknowledge a potential limitation: padded DoH queries might reduce the effectiveness of such classifiers, although this aspect remains unexplored in their evaluation.

Broader Implications. Together, these two studies exemplify the growing trend of applying models from natural language processing and structured learning to network security. They demonstrate that encryption is not a definitive barrier to ML based inference. Rather, it shifts the focus from payload inspection to temporal, statistical, and structural traffic features. In both cases, ML models effectively learn subtle patterns embedded in traffic metadata, which can be used to infer origin, behavior, and even the type of client application in use.

As encrypted communication becomes the norm, the ability to infer semantic information from such data without compromising user privacy is critical. These results suggest that future work should continue exploring self supervised pretraining, cross task generalization, and robustness to evasion techniques such as query padding or traffic morphing, to maintain the relevance and utility of ML driven network analysis systems.

5.2. Anomaly Detection

This subsection explores machine learning techniques developed for identifying anomalous behavior in network traffic, particularly in Internet of Things (IoT) environments. These techniques aim to model normal network activity and subsequently flag deviations that may indicate malicious actions, such as botnet communication, data exfiltration, or command-and-control signaling. The ability to detect anomalies in real time is essential for modern cybersecurity systems, especially as the complexity and scale of networked devices grow. As encrypted protocols become ubiquitous and attackers adopt increasingly stealthy tactics, unsupervised and semi-supervised learning approaches have gained traction due to their adaptability and minimal reliance on labeled data.

Autoencoder-based IoT Anomaly Detection. Meidan et al. [28] proposed N-BaIoT, one of the earliest works targeting anomaly detection for IoT devices. Their framework trains per-device deep autoencoder models on benign traffic to learn compact representations of normal behavior. An autoencoder is composed of two parts: an encoder that compresses the input into a latent space, and a decoder that reconstructs the original input from this representation. In N-BaIoT, any traffic that results in a high reconstruction error is flagged as anomalous. The authors evaluated their model on network traffic infected by Mirai and BASHLITE botnets and achieved a perfect true positive rate with a false positive rate as low as 0.007. This per-device modeling approach ensures that the system is sensitive to subtle deviations unique to each device's behavioral profile. As IoT devices are typically low-power and task-specific, such profiling enables precise detection of malicious deviations while maintaining low inference costs.

Frequency-Domain Statistical Clustering. Fu et al. [47] introduced Whisper, a statistical anomaly detection framework that leverages frequency-domain characteristics of network traffic for robust and low-latency detection. The system extracts frequency and timing features from packet streams using Discrete Fourier Transformation (DFT), transforming the temporal traffic profile into the frequency domain. These frequency features are then clustered, and any test sample that lies beyond a threshold distance from the nearest cluster is labeled as anomalous. Since the model is trained only on benign traffic, it aligns with the principles of semi-supervised learning. In empirical evaluations, Whisper achieved near-instantaneous inference speeds (average latency of 0.0361 seconds) and demonstrated robustness against adversarial injection attacks. For instance, the system maintained effectiveness even when benign-looking packets were mixed into malicious

streams, with a limited 10.46% degradation in performance as measured by the area under the ROC curve. These results suggest that frequency-domain representations can capture meaningful signal structures resilient to evasion techniques.

Discussion and Implications. Figure 5 compares two representative anomaly detection techniques, autoencoder-based and frequency-domain statistical methods, highlighting their respective architectures and detection pipelines. The left side of the figure illustrates the N-BaIoT framework, where an encoder-decoder structure is trained per IoT device to reconstruct benign traffic patterns. Anomalies are identified based on high reconstruction error. The right side depicts Whisper, where network traffic undergoes a frequency transform followed by clustering, and deviations from learned cluster patterns are flagged as anomalies.

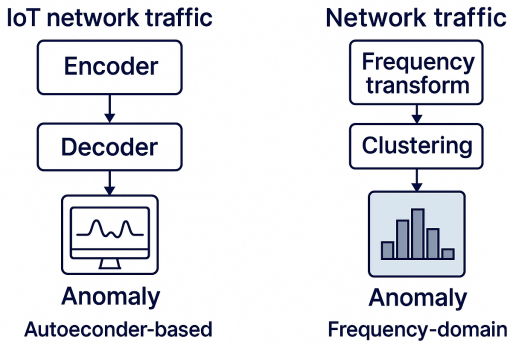

Figure 5. Comparison of two anomaly detection frameworks. The left shows an autoencoder-based approach (e.g., N-BaIoT) that reconstructs benign IoT traffic to detect deviations. The right illustrates a frequency-domain method (e.g., Whisper) that transforms network traffic using Discrete Fourier Transform and clusters the results to flag anomalies.

Both N-BaIoT and Whisper exemplify the viability of unsupervised and semi-supervised approaches in complex, real-world environments. N-BaIoT excels in high-fidelity, device-specific modeling, offering fine-grained detection at low inference cost. Whisper, in contrast, emphasizes generalizability and lightweight deployment by leveraging statistical signal processing in the frequency domain. A key advantage of both is their minimal reliance on labeled data, making them attractive for large-scale cybersecurity deployments where labeled anomalies are scarce.

Despite their strengths, several challenges remain. Deep autoencoders, as black-box models, often lack interpretability, limiting their use in environments where explainability is critical. Moreover, the scalability of per-device training becomes a concern as IoT deployments scale into millions of devices, necessitating distributed or federated solutions. Frequency-domain methods, while efficient, may require careful tuning of signal transformation parameters and thresholds for robust deployment across heterogeneous traffic patterns. Nonetheless, these techniques establish a strong foundation for adaptive threat monitoring, particularly when integrated with rule-based systems or collaborative learning architectures.

5.3. Challenges

Despite significant advancements in applying machine learning to network traffic analysis and anomaly detection, several critical challenges remain that hinder real-world deployment. Chief among these are issues related to interpretability, scalability, and privacy preservation, all of which are essential considerations in high-stakes cybersecurity environments.

One of the most pressing concerns is the interpretability of detection models. Many of the techniques introduced in this survey, such as transformers, deep auto-encoders, and

frequency-domain statistical methods, operate as opaque black boxes. Their decisions are often not accompanied by human-understandable explanations, which makes them difficult to audit, verify, or trust in operational settings. This lack of transparency can be particularly problematic in scenarios such as incident response or forensic analysis, where operators need clear justifications for alerts. Among the reviewed techniques, tree-based models like those in [46] are among the few that offer rule-based interpretability by design.

To address these limitations, recent research has explored explainable AI (XAI) methods tailored to cybersecurity use cases. For instance, Han et al. introduced DeepAID [50], a post-hoc explanation framework for anomaly detection that highlights which input features contributed most to an anomaly score by comparing them against reference profiles of normal behavior. Similarly, Jacobs et al. [49] proposed a decision tree approximation method, which translates the behavior of complex black-box models into simplified and interpretable decision rules. These efforts represent promising steps, yet more systematic solutions are needed to integrate interpretability as a core design principle, rather than an afterthought.

Scalability poses another major challenge in deploying machine learning-based security systems. As network infrastructures grow in size and complexity, especially in the context of Internet of Things (IoT) ecosystems, machine learning models must process high volumes of heterogeneous traffic data in real time. This introduces substantial computational overhead for both training and inference. Traditional centralized learning architectures are often inadequate for such workloads due to limitations in bandwidth, latency, and compute resources.

Furthermore, per-device modeling approaches, such as those used in N-BaIoT [**?**], while effective in capturing fine-grained behavior, become impractical when deployed across networks containing thousands or millions of devices. The training and maintenance of distinct models for each endpoint can quickly lead to unacceptable overhead and resource consumption.

To mitigate these issues, distributed learning paradigms have been explored. Notably, Marfo et al. [51] apply federated learning to the anomaly detection task, enabling collaborative model training across multiple devices while avoiding the need to transmit raw data. This architecture not only enhances scalability but also opens avenues for learning across organizational or jurisdictional boundaries where data sharing may be restricted.

Privacy and regulatory concerns further complicate the adoption of centralized data-driven approaches. Many cybersecurity datasets are sensitive, containing user metadata or potentially identifying information. The centralized aggregation of such data for model training may violate privacy policies or legal regulations such as GDPR or HIPAA, depending on the deployment context. Federated learning and privacy-preserving techniques such as secure aggregation and differential privacy offer potential solutions, though they come with trade-offs in terms of model complexity, convergence rates, and communication costs. Designing learning systems that are both effective and privacy-aware remains a central open problem.

Beyond technical considerations, practical integration into operational cybersecurity environments presents another layer of complexity. Models must be compatible with existing infrastructure, interoperable with security information and event management (SIEM) systems, and resilient to adversarial manipulation. Additionally, black-box models that are difficult to debug or validate pose challenges for compliance and certification in regulated sectors such as healthcare, finance, and energy.

In summary, while machine learning models for encrypted traffic analysis and anomaly detection have demonstrated strong empirical performance, their adoption in real-world environments hinges on addressing broader systems-level concerns. Future research must

aim to develop models that are not only accurate but also interpretable, scalable, privacy-preserving, and operationally robust. Advances in these areas will be critical for translating academic innovations into practical, trustworthy cybersecurity solutions.

6. Adversarial Approaches

Adversarial machine learning refers to the intentional manipulation of inputs to deceive machine learning models into producing incorrect or unintended outputs, such as misclassifications, altered embeddings, or faulty completions. In the context of this survey, adversarial techniques are particularly relevant, as attackers must circumvent or mislead ML-based systems designed to detect their malicious behavior. This section examines two primary categories of adversarial strategies: evasion attacks, which occur during the inference phase by subtly altering inputs, and backdoor or trojan attacks, which compromise the model during training. Additionally, we highlight several studies that do not neatly fall into either category but offer important perspectives on emerging attack vectors and defenses.

6.1. Evasion Attacks

Adversarial evasion refers to the class of attacks that occur during the inference phase of a machine learning model's lifecycle. In such attacks, adversaries deliberately craft inputs known as adversarial examples that cause the model to produce incorrect or misleading outputs. These inputs often involve only minimal perturbations to legitimate data samples, such that they remain functionally equivalent or semantically similar from a human perspective but are misclassified by the model. In the context of cybersecurity, these perturbations may be applied to malware binaries, network traffic traces, or source code artifacts in a way that defeats model-based detection while preserving malicious intent.

A representative example is provided by Rafiq et al. [52] in their work on evading Drebin, a state of the art Android malware classifier. They demonstrate that the injection of merely three semantic preserving features into a malware sample can yield a 100% evasion rate against the original model. However, their study also suggests a promising defense: using an ensemble of classifiers trained on disjoint feature subsets. This ensemble approach significantly reduces susceptibility to the attack, maintaining 91% accuracy even when up to 14 features are perturbed.

Adversarial vulnerabilities are not limited to malware classifiers. Convolutional neural networks, frequently used in vision-based and binary analysis tasks, are also highly susceptible to such attacks. A notable example is the black box boundary attack presented in the literature [53], which requires no access to a model's internal weights or architecture. By interacting only with the model's output predictions, the authors achieve a 95.2% evasion success rate. This demonstrates that even without white box access, adversarial example generation remains highly feasible. Maho et al. [53] further improve on this idea by introducing a geometric method for adversarial perturbation that requires significantly fewer model queries. Their technique does not rely on gradient estimation or surrogate modeling. Instead, it iteratively refines perturbations using the structure of the model's decision boundary. While such techniques are tailored for vision models, their adaptation to binary or network data domains involves different challenges such as preserving executable semantics or protocol correctness. Future research should explore how convolutional models trained on binary code respond to similar adversarial mechanisms, especially given the non-visual nature of such data.

Beyond convolutional networks, graph neural networks, increasingly used for network traffic and structural code analysis, also face evasion risks. In one study [54], researchers investigate defenses against Nettack, a powerful adversarial attack on graph models. They

propose a low rank approximation strategy that simplifies graph structure, thereby reducing the attack surface. Their experiments reveal that low rank approximated graphs are more resilient to adversarial noise, although they also discover that low rank perturbations can still bypass these defenses. Interestingly, such low rank attacks tend to introduce more detectable artifacts, such as anomalies in node degree distributions, offering potential avenues for defensive filtering or anomaly detection.

Semantic preserving adversarial attacks are also demonstrated in high level code analysis models. One line of work introduces DAMP [55], an attack framework targeting source code level models such as code completion, vulnerability detection, and malware classification systems. The authors show that simple code transformations like renaming variables or reordering functions can drastically alter a model's decision, despite leaving program behavior unchanged. With a targeted success rate of 89%, their results emphasize the need for robust defenses in models analyzing syntactically complex input. Furthermore, retraining models with adversarially perturbed examples, along with outlier detection on feature representations such as variable names, significantly improves robustness. This demonstrates the value of adversarial training and semantic filtering in code analysis applications.

In real world deployments, attackers are unlikely to have full access to model parameters or architecture. Under such black box assumptions, adversaries rely on repeated queries to infer model behavior and craft successful attacks. This query based threat model is explored in several works. One proposed framework [56] uses a stateful detection mechanism based on nearest neighbors and autoencoder based similarity metrics to track the distribution of input queries. By measuring the distance between new queries and prior ones, the system flags repetitive or highly similar queries that may indicate ongoing adversarial probing. Researchers report that a large number of suspicious queries are detected well before a successful adversarial example is produced. While this detection approach is particularly effective for black box attacks, the authors also highlight the need for better defenses against white box attacks, which remain a significant open challenge.

Taken together, the research on adversarial evasion highlights the broad vulnerability of machine learning models ranging from binary classification and code analysis to graph-based and vision models. These attacks not only expose blind spots in existing defenses but also reveal important design tradeoffs between robustness, interpretability, and performance. As adversaries continue to exploit the fragility of deep learning systems, future defense strategies must evolve accordingly. Promising directions include ensemble learning, model distillation, input sanitization, adversarial training, and the development of robust representation learning techniques that prioritize semantic fidelity over surface-level features. More importantly, detection mechanisms must account for both black box and white box scenarios, and defensive measures should be tailored to the specific data modality and threat model at hand.

Figure 6 provides a visual summary of several representative adversarial evasion strategies discussed in this subsection. The figure categorizes evasion attacks into three primary scenarios: (1) semantic-preserving feature perturbation in binary code, where carefully crafted modifications are made to binary features without altering program behavior; (2) query-based black-box attacks, in which an adversary interacts with a model through API access to infer vulnerabilities and generate adversarial inputs; and (3) structural perturbation in graph-based models, where modifications to graph topology (e.g., node connections or attributes) are used to evade detection. These attack vectors reflect the diversity of evasion techniques across data modalities and model types, underscoring the need for robust and adaptive defenses.

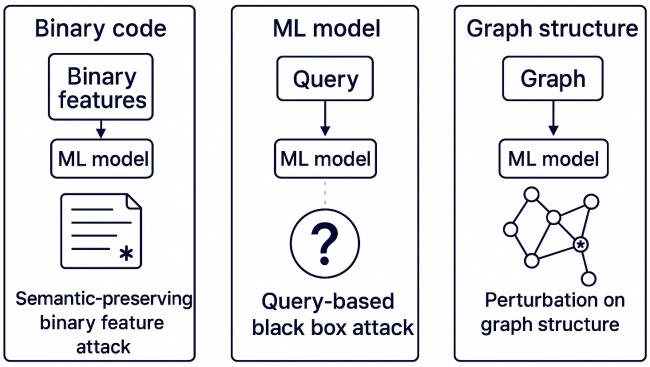

Figure 6. Overview of adversarial evasion techniques targeting ML-based cybersecurity systems. From left to right: (1) Semantic-preserving feature perturbation in binary code, (2) Query-based black-box attacks, and (3) Perturbations on graph-structured data.

6.2. Backdoor and Trojan Attacks

Backdoor or Trojan attacks represent a stealthy form of model poisoning, where the adversary manipulates the model during training so that it behaves normally under standard inputs but misclassifies any input that contains a specific trigger pattern. These attacks are particularly dangerous because the malicious behavior is often undetectable under normal evaluation and can be precisely controlled by the attacker to produce targeted misclassification. In security-critical applications such as malware detection or intrusion prevention, backdoored models pose serious risks to system integrity.

Tang et al. [57] introduce *TrojanNet*, a versatile and stealthy framework for implanting trojans in deep neural networks. Their approach overlays a small trojan subnetwork onto a host model and trains it to respond only to specific trigger patterns that resemble compact QR codes. To enhance stealth, the trigger size is kept minimal (4×4 pixels), and the trojan network is trained on noisy inputs to suppress accidental activations on benign samples. The proposed attack achieves a 100% success rate across various benchmark tasks, demonstrating the feasibility and transferability of such general trojan mechanisms across DNN architectures.

To defend against such threats, Gao et al. [58] propose *STRIP* (Sensitivity to Perturbation), a black box runtime detection method. STRIP leverages the input-agnostic nature of most trigger patterns, meaning the trigger works independently of the actual input content. By adding random perturbations to inputs and observing the entropy of the model's outputs, STRIP is able to differentiate between clean and trojaned samples. Benign models typically show high output diversity (that is, high entropy) under such perturbations, while backdoored models produce consistently low entropy outputs due to the strong influence of the trigger. STRIP is architecture independent and deployable without access to the model internals, making it practical for deployment in real-world systems.

Another defense strategy is input purification, exemplified by *Februus* [60], which aims to remove or neutralize potential triggers from input samples at runtime. Februus identifies the most salient region influencing the model's prediction, masks it, and then reconstructs the image using an autoencoder. This helps eliminate adversarial influence while preserving task-relevant features. The approach significantly reduces the success rate of backdoor attacks across four datasets from 100 percent to under 0.5 percent. Like STRIP, Februus is model agnostic and does not require retraining or access to weights, offering a viable plug and play defense mechanism.

However, both STRIP and Februus share a common limitation. They assume the trojan trigger is input-agnostic and easily separable from benign features. This assumption is challenged by more sophisticated attack strategies. Wang et al. [59] propose *BppAttack*,

an advanced input-dependent trojan attack that generates human imperceptible perturbations to avoid detection. Unlike prior approaches that use static triggers, BppAttack crafts unique perturbations tailored to each input image, thus avoiding entropy-based detection mechanisms like STRIP and resisting region-based defenses like Februus. Moreover, BppAttack does not require training an auxiliary trojan model, simplifying the attack process. The authors show that this method successfully evades several state-of-the-art defenses, including Grad-CAM and entropy-based filters.

These studies illustrate the escalating complexity in the arms race between adversarial attacks and defense strategies. While early backdoor attacks exploited static and easily detectable triggers, modern attacks have evolved to adopt dynamic input-sensitive patterns that evade traditional detection. At the same time, defenders have begun integrating robust runtime detection, semantic purification, and statistical analysis to counter emerging threats. However, as attacks grow more sophisticated, so must the defenses.

In practice, securing ML systems against Trojan threats will require a combination of strategies, including adversarial training, input sanitization, anomaly detection, and regular model audits. Moreover, continued research is needed to develop universally effective defenses that function in black box environments and remain robust under both input-agnostic and input-aware attack scenarios. Given the rising deployment of AI models in high-stakes applications, proactive defense against trojan attacks is not optional but essential.

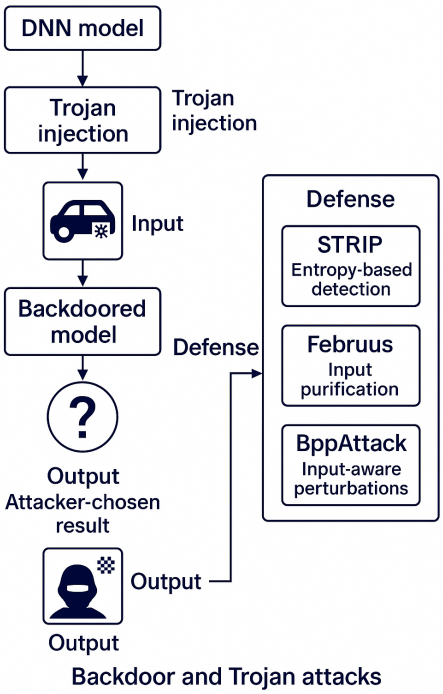

Figure 7. Illustration of backdoor and trojan attack processes in ML systems, including training-phase poisoning and inference-time misclassification triggered by stealthy input patterns. Defensive strategies such as STRIP and Februus are also highlighted.

Figure 7 provides a visual summary of typical backdoor and trojan attack scenarios in adversarial machine learning. The figure highlights three main components of this threat landscape: (1) the attacker inserts a trigger into selected inputs during the training phase, (2) the compromised model learns to associate the trigger with an attacker-specified class, and (3) during inference, the presence of the trigger reliably causes a misclassification, while clean inputs are correctly classified. This illustrates both the stealth and potency of such attacks. Additionally, the figure includes defense mechanisms such as entropy-based

detection (e.g., STRIP) and input purification (e.g., Februus), which attempt to mitigate these threats by analyzing model behavior at runtime or modifying input data to neutralize potential triggers. Together, these elements reflect the cat-and-mouse dynamics between attack strategies and defensive countermeasures in trojaned ML systems.

6.3. Data Poisoning

Data poisoning attacks target the integrity of machine learning models by manipulating the training data, thereby embedding harmful behaviors or degrading predictive performance. Unlike evasion attacks that occur at inference time, poisoning fundamentally alters model parameters during training, making it particularly damaging in long-term learning systems and pipelines with continuous updates. Early foundational work by Biggio et al. [1] demonstrated that carefully crafted samples can corrupt classical machine learning algorithms such as support vector machines. Subsequent studies extended these concepts to deep neural networks, showing that poisoning attacks can stealthily introduce targeted misclassification behaviors without significantly affecting model accuracy on clean validation data.

Poisoning attacks are commonly categorized into *dirty-label* and *clean-label* variants. Dirty-label attacks assume the adversary can tamper with labels, such as flipping benign samples to malicious classes or inserting mislabeled training points. Mei and Zhu [2] viewed poisoning through the lens of machine teaching, computing optimal poisoning samples to efficiently steer the learner. In contrast, clean-label poisoning assumes that labels remain correct, forcing the attacker to modify only the features. Clean-label attacks are highly stealthy because they resemble legitimate data, yet can induce feature collisions or decision-boundary shifts. Shafahi et al. [4] introduced clean-label "Poison Frogs," showing that feature-space alignment alone can cause targeted, hard-to-detect misclassification in deep networks. Turner et al. [5] further showed that clean-label poisoning can embed subtle "triggers" mirroring backdoor behaviors while keeping ground-truth labels unchanged.

Cybersecurity presents unique challenges for poisoning defenses. Real-world security data sources, including network telemetry, malware corpora, and IDS logs, often originate from untrusted endpoints or distributed monitoring agents. Thus adversaries may subtly inject malicious artifacts during data collection or labeling. Steinhardt et al. [3] demonstrated that seemingly minor perturbations can shift decision boundaries with surprisingly small poisoning budgets, raising concerns for large-scale security monitoring systems. Furthermore, threat intelligence feeds and malware-sharing platforms create opportunities for sophisticated adversaries to seed poisoning samples intentionally or inadvertently, particularly in automated retraining environments.

Defensive methods aim to detect and filter malicious data or ensure model robustness against such contamination. Influence functions [6] attempt to identify training points disproportionately impacting model predictions, flagging potential poisoning samples. Data sanitization and clustering techniques can remove outliers or identify anomalous feature distributions prior to training. In distributed learning scenarios, such as federated learning for security analytics, Byzantine-resilient aggregation methods such as Krum and Bulyan [7,8] filter malicious client gradients by rejecting outliers during model update phases. Differential privacy methods [9] introduce noise during training, making precise gradient manipulation more difficult, though at the cost of utility in high-dimensional security tasks.

Despite these advances, defending against poisoning remains difficult due to high data heterogeneity and adversary adaptability. Clean-label attacks are particularly insidious in malware and network intrusion detection tasks, where natural variability in binary features or traffic behaviors complicates anomaly-based filtering. Moreover, systems that

continuously retrain models on streaming security data amplify the risk of long-term poisoning accumulation. Future directions include robust self-supervised security learning pipelines, certifiably robust training for streaming datasets, and dynamic trust scoring for decentralized data contributors. In summary, poisoning represents a critical and evolving threat to machine learning-enabled cybersecurity, particularly as automated pipelines and online learning become more prevalent.

6.4. Model Extraction and Privacy Attacks

Model extraction and privacy attacks exploit model access interfaces to replicate proprietary models or infer training data properties. Tramèr et al. [10] first demonstrated that commercial machine learning APIs are vulnerable to extraction via adaptive querying, allowing adversaries to approximate decision boundaries and steal models with high fidelity. These attacks undermine intellectual property protections, enabling adversaries to duplicate expensive security models such as malware classifiers or phishing detectors. Extraction also provides a precursor to further attacks, as a cloned model can be used offline to craft targeted adversarial examples.

Beyond functional replication, privacy attacks reveal sensitive information contained in training data. Early work by Fredrikson et al. [13] showed that logistic regression models could leak sensitive patient attributes via inversion. Shokri et al. [11] formalized membership inference attacks (MIAs), demonstrating that adversaries can determine whether specific samples were used during training. For cybersecurity models trained on sensitive enterprise logs, traffic telemetry, or malware signatures, such leakage may reveal internal threat data or proprietary defense behaviors. Recent work by Carlini et al. [12] showed that MIAs can be derived from first principles and extended to modern deep networks, reinforcing that privacy risk persists across architectures and domains.

Real-world cybersecurity systems magnify these risks because models are often deployed in cloud services, SOC pipelines, and federated monitoring architectures. Attackers may access security models via threat intelligence APIs, endpoint detection and response (EDR) platforms, vendor dashboards, or packet inspection appliances. Even limited query feedback, such as scores or confidence rankings, can accelerate extraction or MIA feasibility. Meanwhile, federated and collaborative frameworks expose gradient pathways that may disclose training data through gradient leakage [14]. Cybersecurity telemetry, being highly sensitive and proprietary, amplifies the consequences of such leakage.

Defense mechanisms span interface restrictions, privacy-preserving learning, and cryptographic safeguards. Confidence masking, quantization, and randomized response degrade information leakage but may weaken detection accuracy or analyst trust in outputs. Differential privacy [9] offers strong formal guarantees, though utility degradation remains non-trivial for complex security tasks. Secure aggregation protocols [14] and federated learning frameworks [15] reduce exposure of raw gradients, yet remain vulnerable to adaptive and collusion-based threats. Meanwhile, rate limiting, perturbed access logging, and watermarking of model outputs aim to identify or deter extraction attempts in MLaaS settings.

Despite progress, achieving strong confidentiality for ML-based security systems remains unsolved. Cyber defense models must process high-volume, high-sensitivity telemetry while providing actionable results with low latency. This creates a trade-off: stronger privacy constraints increase resilience but can delay threat detection or reduce fidelity. Future research must address robust privacy-preserving architectures that scale to streaming real-time detection, integrate secure enclaves and hardware security modules, and provide certified robustness against extraction and inference attacks. As security functions increasingly rely on shared intelligence, cross-organization collaboration, and

federated defense, preserving model confidentiality and training data privacy will remain central to trustworthy ML-enabled cybersecurity.

6.5. Evolving Challenges and Defense Strategies

The domain of adversarial attacks is evolving rapidly, with increasingly sophisticated methods targeting machine learning systems. For example, STRIP was initially proposed as a state of the art runtime defense against trojan attacks, but was later outmaneuvered by the BppAttack [57,59], which introduced human imperceptible, input dependent perturbations that evade entropy based detection strategies. These developments reflect a growing trend toward generalized frameworks for both attack and defense, emphasizing architectural independence and adaptability.

However, for cybersecurity practitioners, generalized solutions alone may not provide sufficient confidence in the robustness of deployed models. Effective security requires tailoring defenses to specific model architectures and application contexts. As discussed in the malware detection section, implementation choices such as the use of convolutional models, transformers, or graph neural networks significantly influence a model's susceptibility to adversarial manipulations. Practitioners must possess not only a general understanding of adversarial tactics but also deeper, context specific expertise to deploy resilient systems. Tools such as the stateful detector proposed in [56] offer interpretable, runtime metrics to identify suspicious query behaviors and inform dynamic security policies.

The rise of large language models introduces additional complexities. Derner et al. [61] present a comprehensive taxonomy of malicious uses of LLMs, highlighting risks such as automated malware generation, phishing content synthesis, and model misuse for social engineering. These capabilities lower the technical threshold for attackers, expanding the pool of potential threat actors. As LLM architectures such as GPT evolve over time, their exploitable behaviors also change, creating a moving target for defenders.

In this context, cybersecurity professionals must extend their threat modeling expertise to LLM based systems. Ensuring robust safeguards such as fine tuned guardrails, adversarial robustness evaluation, and responsible model deployment practices is essential. Moreover, continuous monitoring and red teaming are necessary to track and respond to emerging attack vectors as models are updated or refined.

In summary, adversarial machine learning continues to present evolving challenges and opportunities. Achieving real world security for machine learning applications demands a combination of general awareness, domain specific insight, and proactive collaboration between cybersecurity and artificial intelligence communities.

7. Future Research Directions

Despite impressive progress across malware detection, network anomaly detection, and adversarial machine learning, many critical challenges remain unresolved. The evolving threat landscape, growing complexity of deployed machine learning models, and increasing regulatory demands call for a deeper investigation into methods that are not only accurate but also secure, interpretable, and scalable. Based on the comprehensive analysis presented in this survey, we outline several promising directions for future research.

Few-Shot and Zero-Shot Malware Detection: A recurring limitation identified across malware detection studies is their reliance on many-shot classification settings. However, real-world environments often feature new or rare malware families for which labeled data is sparse or unavailable. Developing ML systems capable of few-shot or zero-shot generalization, potentially through meta-learning, prototype networks, or contrastive learning, remains an open challenge. Incorporating self-supervised pretraining objectives,

such as control-flow prediction or semantic embedding alignment, may further enhance generalizability to novel threats.

Transfer Learning Across Modalities: While transfer learning has proven effective in natural language and vision domains, its applicability to low-level code analysis, binary classification, and network security is still underexplored. Questions remain regarding how pretrained representations from domains like NLP or CV can be adapted to code semantics or encrypted traffic flows. For example, can models trained on source code reliably enhance binary-level classifiers? Investigating transferability between textual, visual, and binary modalities could improve cross-domain robustness and reduce reliance on task-specific datasets.

Standardized Evaluation of Adversarial Robustness: The lack of standardized benchmarks for evaluating robustness against adversarial attacks is a persistent gap. Unlike in computer vision or NLP, the adversarial threat landscape in cybersecurity varies significantly by modality and attacker capability (e.g., binary manipulation vs. API query attacks). Future work should develop unified evaluation frameworks, robust metrics, and threat models that reflect the practical adversarial risks encountered in malware detection, anomaly detection, and traffic analysis. Establishing such standards would also aid in comparative studies of defense methods.

Interpretable and Trustworthy Detection Models: Interpretability remains a major barrier to the adoption of deep learning in operational cybersecurity environments. Black-box models like transformers and autoencoders often lack mechanisms for explaining predictions in human-readable form. This limitation impedes forensic analysis, security policy tuning, and compliance reporting. Advancing post-hoc explanation methods (e.g., DeepAID, model distillation) and developing inherently interpretable models (e.g., decision tree surrogates, attention visualization) is a crucial area for future work. Large language models also offer new opportunities in this space: their generative capabilities could be used to translate model outputs into contextualized, natural language justifications.

Federated Learning for Privacy-Preserving Security: As IoT deployments grow, so too do concerns about privacy, bandwidth, and data ownership. Federated learning offers a promising approach to collaboratively train models without centralized data aggregation. However, applying FL in security contexts introduces new challenges, including heterogeneous data distributions, variable device capabilities, and adversarial participation. Research is needed to develop robust and communication-efficient FL algorithms tailored for security-critical domains, and to understand the trade-offs between local specialization and global generalization in anomaly detection or malware classification tasks.

Balancing Model Complexity and Attack Resilience: Several studies have noted that increased model complexity (e.g., deeper networks or high-rank GNNs) can make systems more vulnerable to adversarial attacks. Conversely, models approximated with lower complexity (e.g., low-rank graph approximations) may offer improved robustness at the cost of expressiveness. Future research should explore this trade-off more systematically. What types of architectural simplification actually improve security? Can ensembles of simpler models outperform complex monoliths in adversarial settings? Quantifying the relationship between model sophistication and attacker effort could help guide defensive design.

Securing Foundation Models and LLMs in Cybersecurity: Large foundation models, particularly LLMs, are increasingly being integrated into cybersecurity workflows-for example, to assist in vulnerability triage, generate security reports, or automate detection rules. However, their general-purpose nature and evolving behavior present unique security risks. Malicious actors may exploit LLMs for code generation, phishing, or prompt injection. Future work must explore how to harden LLMs against misuse, including red

teaming, adversarial prompting detection, and fine-tuning strategies that embed security domain knowledge. Additionally, LLMs could be leveraged as tools to interpret opaque ML decisions or assist in crafting human-readable security explanations.

Toward Holistic, Multi-Layered Defense Architectures: Finally, future work should focus on integrating diverse ML models into layered defense architectures that combine anomaly detection, signature-based systems, and runtime interpretability mechanisms. Such hybrid systems could incorporate statistical detectors (e.g., Whisper), transformer-based classifiers, and query anomaly monitors (e.g., stateful detection) in a coordinated framework. This approach would allow defenses to span different temporal and semantic resolutions, improving robustness against multi-phase and polymorphic attacks.

Overall, future research at the intersection of machine learning and cybersecurity must expand beyond improving classification accuracy. It must address practical deployment constraints, regulatory concerns, and adversarial resilience. By developing interpretable, generalizable, and secure ML systems, researchers can ensure that the benefits of intelligent security technologies are realized without compromising trust, privacy, or robustness in critical applications.

8. Conclusions

The intersection of machine learning and cybersecurity represents a dynamic and rapidly evolving research frontier. As this survey has shown, machine learning techniques are increasingly central to both defensive and offensive applications in cybersecurity, powering tasks such as malware detection, anomaly detection, and traffic classification, while also introducing new attack surfaces through adversarial manipulation. The dual use nature of machine learning presents both immense opportunities and complex challenges for securing modern digital infrastructures.

Throughout this paper, we have examined a wide range of machine learning models and algorithms used across various security domains, highlighting their strengths, limitations, and underlying assumptions. Special attention was given to adversarial machine learning, including evasion, poisoning, and backdoor attacks, which expose critical vulnerabilities in model behavior and training pipelines. We also discussed privacy risks such as membership inference and model extraction, particularly in distributed and federated learning environments.

In addition to reviewing technical advances, this survey identified key challenges related to model interpretability, scalability, data availability, and privacy. These challenges remain open areas for future research and have practical implications for real world deployment. While recent efforts have made progress, such as the development of interpretable anomaly detectors, secure federated frameworks, and robust adversarial defenses, there is a pressing need for solutions that balance performance with trustworthiness, transparency, and ethical responsibility.

Ultimately, our goal is to provide researchers, practitioners, and students with a comprehensive and accessible overview of how machine learning is shaping the future of cybersecurity. By synthesizing both foundational concepts and recent advances, this survey aims to support the development of more resilient, intelligent, and secure systems in an increasingly connected world.

Acknowledgments: Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-24-1-0088. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec)*, 2013, pp. 143–152, doi: 10.1145/2517312.2517315.
2. S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015, pp. 2871–2877.
3. J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
4. A. Shafahi, W. Huang, M. Najibi *et al.*, "Poison Frogs! Targeted clean-label poisoning attacks on neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
5. A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," arXiv:1912.02771, 2019.
6. P. W. W. Koh and P. S. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, pp. 1885–1894.
7. P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
8. E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.
9. M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 308–318, doi: 10.1145/2976749.2978318.
10. F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *USENIX Security Symposium*, 2016, pp. 601–618.
11. R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy (S&P)*, 2017, pp. 3–18, doi: 10.1109/SP.2017.41.
12. N. Carlini, F. Tramèr, E. Wallace *et al.*, "Membership inference attacks from first principles," in *IEEE Symposium on Security and Privacy (S&P)*, 2022, pp. 1897–1914, doi: 10.1109/SP46214.2022.9833668.
13. M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.
14. K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191, doi: 10.1145/3133956.3133982.
15. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.
16. A. Zarras, P. Bountakasa, A. Lekidisa, and C. Xenakisa, "Defense Strategies for Adversarial Machine Learning: A Survey," *Computers & Security Review*, vol. 48, no. 1, p. 100573, 2023, doi: 10.1016/j.cosrev.2023.100573.
17. S. Pelekis, T. Koutroubas, A. Blika, A. Berdelis, E. Karakolis, C. Ntanos, E. Spiliotis, and D. Askounis, "Adversarial Machine Learning: A Review of Methods, Tools, and Critical Industry Sectors," *Artificial Intelligence Review*, vol. 58, no. 2, article 226, 2025, doi: 10.1007/s10462-025-11147-4.
18. A. Vassilev, A. Oprea, A. Fordyce, and H. Anderson, "Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations," NIST AI 100-2e2023, National Institute of Standards and Technology, 2024. Available: https://doi.org/10.6028/NIST.AI.100-2e2023.
19. L. Demetrio, S. E. Coull, B. Biggio, G. Lagorio, A. Armando, and F. Roli, "Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection," *arXiv preprint arXiv:2008.07125*, 2020.
20. S. Seneviratne, R. Shariffdeen, S. Rasnayaka, and N. Kasthuriarachchi, "Self-Supervised Vision Transformers for Malware Detection," *arXiv preprint arXiv:2208.07049*, 2022.
21. M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Membership Inference Attacks and Defenses," *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

22. H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership Inference Attacks on Machine Learning: A Survey," *ACM Computing Surveys*, 2022.
23. Li Bai, Haibo Hu, Qingqing Ye, Haoyang Li, Leixia Wang, and Jianliang Xu, "Membership Inference Attacks and Defenses in Federated Learning: A Survey," arXiv preprint arXiv:2412.06157, 2024.
24. L. Bai, H. Hu, Q. Ye, H. Li, L. Wang, and J. Xu, "Membership Inference Attacks and Defenses in Federated Learning: A Survey," *arXiv preprint arXiv:2412.06157*, Dec. 2024.
25. Wu, Hengyu and Cao, Yang, "Membership inference attacks on large-scale models: A survey," *arXiv preprint arXiv:2503.19338*, 2025.
26. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.
27. A. Rahali and M. A. Akhloufi, "MalBERT: Using Transformers for Cybersecurity and Malicious Software Detection," *arXiv preprint arXiv:2103.03806*, 2021.
28. Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici, "N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.
29. D. Yumlembam, S. Patel, and R. Kumar, "IoT Anomaly Detection through Deep Neural Networks," *Sensors*, vol. 22, no. 12, 2022.
30. H. Li, J. Zhou, and K. Wang, "PHCNN: Payload-Aware CNN for Malware Classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 1234–1248, 2024.
31. F. Ullah, A. Alsirhani, M. M. Alshahrani, A. Alomari, H. Naeem, and S. A. Shah, "Explainable Malware Detection System Using Transformers-Based Transfer Learning and Multi-Model Visual Representation," *Sensors*, vol. 22, no. 18, p. 6766, 2022.
32. H. Rafiq, "Android Malware Detection Using Machine Learning to Mitigate Adversarial Evasion Attacks," Ph.D. dissertation, Northumbria University, 2022.
33. A. Singla, A. Sukharevsky, L. Yee, and M. Chui, "The state of AI in early 2024: Gen AI adoption spikes and starts to generate value," McKinsey & Company, May 30, 2024. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai
34. J. Schneider, "Malware history," IBM Think, Nov. 25, 2024. https://www.ibm.com/think/topics/malwa history
35. H. Wang et al., "jTrans: jump-aware transformer for binary code similarity detection," Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2022), Jul. 2022, doi: 10.1145/3533767.3534367.
36. X. Li, Y. Qu, and H. Yin, "PalmTree: Learning an Assembly Language Model for instruction embedding," Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Nov. 2021, doi: 10.1145/3460120.3484587.
37. X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification," Proceedings of the ACM Web Conference 2022, Apr. 2022, doi: 10.1145/3485447.3512217.
38. H. Long, Z. Tian, and Y. Liu, "Detecting Android Malware Based on Dynamic Feature Sequence and Attention Mechanism," 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP), pp. 129–133, Jan. 2021, doi: 10.1109/csp51677.2021.9357569.
39. Y. Chai, L. Du, J. Qiu, L. Yin, and Z. Tian, "Dynamic Prototype Network based on Sample Adaptation for Few-Shot Malware Detection," IEEE Transactions on Knowledge and Data Engineering, p. 1, Jan. 2022, doi: 10.1109/tkde.2022.3142820.
40. H. Li, J. Wu, and F. Gu, "PH-CNN for PE Malware Classification by Means of Enhanced Images," 2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR), pp. 104–109, Mar. 2024, doi: 10.1109/isssr61934.2024.00019.
41. C.-D. Nguyen, N. H. Khoa, K. N.-D. Doan, and N. T. Cam, "Android Malware category and family classification using static analysis," 2022 International Conference on Information Networking (ICOIN), Jan. 2023, doi: 10.1109/icoin56518.2023.10049039.
42. O. Aslan and A. A. Yilmaz, "A new malware classification framework based on deep learning algorithms," IEEE Access, vol. 9, pp. 87936–87951, Jan. 2021, doi: 10.1109/access.2021.3089586.
43. R. Elnaggar, L. Servadei, S. Mathur, R. Wille, W. Ecker, and K. Chakrabarty, "Accurate and robust malware detection: running XGBoost on runtime data from performance counters," IEEE

Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 7, pp. 2066–2079, Aug. 2021, doi: 10.1109/tcad.2021.3102007.
44. R. Yumlembam, B. Issac, S. M. Jacob, and L. Yang, "IoT-Based Android Malware detection using Graph neural network with adversarial defense," IEEE Internet of Things Journal, vol. 10, no. 10, pp. 8432–8444, Jul. 2022, doi: 10.1109/jiot.2022.3188583.
45. M. E. Eren, M. Bhattarai, K. Rasmussen, B. S. Alexandrov, and C. Nicholas, "MalwareDNA: Simultaneous Classification of Malware, Malware Families, and Novel Malware," 2023 IEEE International Conference on Intelligence and Security Informatics (ISI), Charlotte, NC, USA, vol. abs/2006.09271, pp. 1–3, Oct. 2023, doi: 10.1109/isi58743.2023.10297217.
46. D. Vekshin, K. Hynek, and T. Cejka, "DOH Insight," Proceedings of the 17th International Conference on Availability, Reliability and Security, Aug. 2020, doi: 10.1145/3407023.3409192.
47. C. Fu, Q. Li, M. Shen, and K. Xu, "Realtime robust malicious traffic detection via frequency domain analysis," Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 3431–3446, Nov. 2021, doi: 10.1145/3460120.3484585.
48. A. Sejfia and M. Schäfer, "Practical automated detection of malicious npm packages," Proceedings of the 44th International Conference on Software Engineering, May 2022, doi: 10.1145/3510003.3510104.
49. A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, "AI/ML for Network Security," Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 1537–1551, Nov. 2022, doi: 10.1145/3548606.3560609.
50. D. Han et al., "DEEPAID: Interpreting and improving deep learning-based anomaly detection in security applications," Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 3197–3217, Nov. 2021, doi: 10.1145/3460120.3484589.
51. W. Marfo, D. K. Tosh, and S. V. Moore, "Network anomaly detection using federated learning," MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM), pp. 484–489, Nov. 2022, doi: 10.1109/milcom55135.2022.10017793.
52. H. Rafiq, N. Aslam, U. Ahmed, and J. C.-W. Lin, "Mitigating malicious adversaries evasion attacks in industrial internet of things," IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 960–968, Jul. 2022, doi: 10.1109/tii.2022.3189046.
53. F. Alrasheedi and X. Zhong, "Imperceptible Adversarial Attack on Deep Neural Networks from Image Boundary," arXiv.org, Aug. 29, 2023. https://arxiv.org/abs/2308.15344
54. N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. E. Papalexakis, "All You Need Is Low (Rank)," WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 169–177, Jan. 2020, doi: 10.1145/3336191.3371789.
55. N. Yefet, U. Alon, and E. Yahav, "Adversarial examples for models of code," Proceedings of the ACM on Programming Languages, vol. 4, no. OOPSLA, pp. 1–30, Nov. 2020, doi: 10.1145/3428230.
56. S. Chen, N. Carlini, and D. Wagner, "Stateful Detection of Black-Box Adversarial Attacks," SPAI '20: Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence, pp. 30–39, Oct. 2020, doi: 10.1145/3385003.3410925.
57. R. Tang, M. Du, N. Liu, F. Yang, and X. Hu, "An Embarrassingly Simple Approach for Trojan Attack in Deep Neural Networks," KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Aug. 2020, doi: 10.1145/3394486.3403064.
58. Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP," ACSAC '19: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 113–125, Nov. 2019, doi: 10.1145/3359789.3359790.
59. Z. Wang, J. Zhai, and S. Ma, "BppAttack: Stealthy and Efficient Trojan Attacks against Deep Neural Networks via Image Quantization and Contrastive Adversarial Learning," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15054–15063, Jun. 2022, doi: 10.1109/cvpr52688.2022.01465.
60. B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "FEBRUUS: Input Purification Defense against Trojan attacks on deep neural network systems," Annual Computer Security Applications Conference, Dec. 2020, doi: 10.1145/3427228.3427264.

61. E. Derner, K. Batistič, J. Zahálka, and R. Babuška, "A Security Risk Taxonomy for Prompt-Based Interaction with Large Language Models," IEEE Access, vol. 12, pp. 126176–126187, Jan. 2024, doi: 10.1109/access.2024.3450388.
62. J. Zhao, R. Masood, and S. Seneviratne, "A review of Computer vision methods in network Security," IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1838–1878, Jan. 2021, doi: 10.1109/comst.2021.3086475.
63. H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership Inference Attacks on Machine Learning: A survey," ACM Computing Surveys, vol. 54, no. 11s, pp. 1–37, Jan. 2022, doi: 10.1145/3523273.