Emerging Trends in AI-Powered Malware Detection: A Review of Real-Time and Adversarially Resilient Techniques

Bhagwant Singh[*], Sikander Singh Cheema[2]

[1]*Department of Computer Science and Engineering, Punjabi University Patiala, India*

[2]*Department of Computer Science and Engineering, Punjabi University Patiala, India*

Abstract: - The rapid evolution of digital threats requires advanced methodologies in malware detection. Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and Blockchain (BC) have emerged as pivotal technologies in this domain. This review delves into the state-of-the-art trends and techniques in AI powered malware detection systems, mainly focusing on their real-time applications and resilience against adversarial attacks. By in depth analysis diverse algorithms and frameworks, we highlight the significant advantages of AI, including improved detection rates, privacy and the capability to adapt to new malware variants. This study's findings suggest that while classical signature based detection methods are now defeated by robust and obfuscation techniques, AI powered systems can effectively identify patterns and anomalies by leveraging vast amounts of data. Additionally, this study explores the role of explainable AI in providing transparency and interpretability, which are essential for building user trust and ensuring the reliability of automated decisions. The review consolidates key insights from recent literature, emphasizing innovative approaches that bolster the robustness of detection systems against sophisticated evasion techniques. By mapping the landscape of AI powered malware detection, this study aims to guide future research and promote the development of more resilient cybersecurity solutions capable of withstanding the challenges posed by increasingly sophisticated malware-attacks.

Keywords*:* Artificial Intelligence, Malware Detection, Real Time Systems, Adversarial Resilience, Cyber Security Solutions.

1. Introduction

In today's digital landscape, the increase in malware presents significant challenges to cybersecurity, affecting individuals, corporations, and government organizations [1][2][3][4]. The rise of advanced attack methods requires new approaches to malware detection and prevention. Traditional signature based methods are becoming less effective due to the rapid evolution and obfuscation techniques used by modern malware. As a result, integrating Artificial Intelligence (AI), Machine Learning (ML), and Deep learning (DL) technologies into malware detection systems has become a focus in the research community[5][6][7]. This introduction aims to outline the emerging trends in AI powered malware detection, emphasizing the need for real-time and resilient systems. The growth of cyber threats, including more sophisticated and elusive malware variants, requires a proactive and adaptive cybersecurity approach. According to the Cybersecurity and Infrastructure Security Agency (CISA), malware attacks have increased by over 300% in recent years, showing a clear trend towards more aggressive and complex cyber threats[8][9]. Malware not only disrupts operations but also poses risks to sensitive data, financial stability, and national security. The emergence of Ransomware has further complicated the landscape, as it has become a prevalent form of malware that can cripple organizations and demand significant ransoms[10][11]. Due to the dynamic nature of malware development, relying solely on static detection mechanisms is no longer adequate. Modern malware often employs techniques such as polymorphism and metamorphism to evade detection. These evolving tactics require adaptive systems that can quickly respond to new threats while maintaining high accuracy and low false positive rates. Artificial Intelligence technologies, particularly machine learning (ML) and deep learning (DL), have shown great promise in enhancing malware detection capabilities[12][13]. Machine learning algorithms can analyze large datasets, learn from patterns, and make predictions based on historical data. Deep learning, a subset of ML, uses artificial neural networks to capture complex feature representations, making it particularly effective for high dimensional data such as network traffic and executable files. The adoption of AI driven methodologies in malware detection allows for the identification of previously unseen malware variants, addressing the limitations of traditional signature based approaches. For example, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been used to classify malware based on their byte code and behavior, respectively[14][15]. This ability to generalize from training data and identify anomalies has significantly improved detection rates in real-time scenarios. In an era where cyber threats can evolve within seconds, the ability to detect malware in real time is crucial. Real-time detection involves not only identifying threats as they occur but also minimizing the time taken to respond to them. Recent advancements in AI driven detection systems have led to the development of architectures capable of processing vast amounts of data rapidly and efficiently. These systems can analyze network traffic, system calls, and user behavior in real time to flag potential threats. Frameworks such as Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) have been enhanced with AI capabilities to facilitate swift and accurate detection of malware[16][17]. For example, using ensemble learning methods, systems can combine multiple classifiers to improve overall performance, increasing the likelihood of catching even the most subtle threats. The effectiveness of real-time detection mechanisms relies heavily on the continuous training of models with up-to-date data to recognize emerging patterns and adapt to new attack vectors. As AI driven malware detection systems gain traction, adversaries are also developing strategies to exploit weaknesses in these technologies. Adversarial machine learning is an emerging field that focuses on understanding how malicious actors can manipulate AI models to evade detection. To counter these tactics, researchers are exploring adversarial training and robust model architectures to enhance resilience against such attacks[18][19]. Adversarial training involves augmenting training datasets with adversarial examples, enabling models to learn to identify and correctly classify perturbed inputs. Furthermore, hybrid models that combine traditional and AI based methods are being proposed to improve robustness and reliability, particularly in adversarial environments. As AI systems become integral to malware detection, the importance of transparency and interpretability cannot be overstated.

1.1 Background and Motivation

The field of cybersecurity has undergone significant changes in recent years, largely due to the continuous evolution of malware. As cybercriminals refine their techniques, the spread of advanced malware variants presents unprecedented challenges for both organizations and individuals[20][21]. Malware has evolved from simple viruses to complex, multifaceted threats, such as Ransomware, spyware, and fileless malware, each designed to exploit system vulnerabilities and evade traditional detection methods. This increasing complexity has led to a significant increase in data breaches, financial losses, and reputational damage, highlighting the urgent need for robust and adaptive security measures. In response to these challenges, the incorporation of artificial intelligence (AI), particularly deep learning, has emerged as a pivotal advancement in malware detection capabilities[22]. Unlike traditional methods that heavily rely on signature based detection, AI algorithms can learn from extensive datasets, identifying subtle patterns and anomalies indicative of malicious activity. This adaptability to new and evolving threats significantly enhances the accuracy and speed of detection, allowing security systems to keep pace with the dynamic nature of cyber threats. Furthermore, the demand for real time detection has never been greater. Organizations operate in environments where swift response to security incidents is critical to mitigating potential damage. Simultaneously, the need for adversarial resilience has emerged as a critical requirement for modern security systems. As attackers increasingly use adversarial techniques to bypass detection mechanisms, it becomes essential for malware detection systems to not only identify threats but also to withstand and adapt to such sophisticated evasion tactics[23]. Thus, the focus on developing AI driven malware detection systems that prioritize real-time response and adversarial resilience represents a crucial frontier in the ongoing battle against cyber threats.

1.2 Scope of the Review

This review provides an examination of the intersection between Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) in the field of malware detection. The focus is on the urgent need for real-time and adversarial resilient systems as cyber threats become increasingly sophisticated and voluminous, outpacing conventional detection methods. The integration of advanced AI techniques is deemed necessary to address this challenge. The review explores a variety of AI driven methodologies that enhance detection efficacy and ensure rapid response times. It delves into different deep learning architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and ensemble methods, evaluating their performance in identifying both known and unknown malware strains. Additionally, real-time detection strategies are investigated for their crucial role in mitigating threats before they can cause harm. A significant aspect of the review involves examining adversarial defenses that protect AI systems against targeted attacks, thereby ensuring their reliability in diverse operating environments. Key questions addressed in this paper include: What are the most effective AI and deep learning techniques for real-time malware detection? How can these systems be designed to withstand adversarial attacks? What role does explainable AI play in fostering trust and interpretability in automated detection mechanisms? By exploring these themes, this review aims to provide a comprehensive perspective on current advancements and future directions in AI driven malware detection.

1.3 Structure of the Paper

This paper is structured to review emerging trends in AI powered malware detection, focusing on techniques for real-time and adversarial resilient systems. It begins with an Introduction, establishing the growing threat of advanced malware and the role of AI in enhancing detection capabilities. Core AI Techniques in Malware Detection then examines specific AI and deep learning models, such as CNNs and GANs, utilized to identify and classify malicious patterns. The section on Real-time Malware Detection Approaches highlights methods designed for rapid threat detection and response in dynamic environments. Adversarial Resilience in Malware Detection explores techniques for countering evasion tactics and securing detection models against adversarial attacks. A Comparative Analysis of Real-time and Adversarial Resilient Techniques provides insights into the effectiveness, challenges, and adaptability of these methods. Emerging Challenges and Future Directions discusses key obstacles, including dataset limitations and the need for transparency in AI driven detection. The paper concludes with a Summary and Future Scope, encapsulating key findings and proposing future research directions for enhancing AI driven malware detection systems.

2. Artificial Intelligence and Deep Learning Techniques for Malware Detection

The rapid increase in malware presents substantial challenges to cybersecurity, necessitating innovative and reliable detection methods. Traditional malware detection systems, which heavily rely on signature based approaches, are becoming less effective against emerging threats. In response, the industry has shifted its focus to Artificial Intelligence (AI) and machine learning techniques, which offer the agility and adaptability required to combat complex malware.

1. Machine Learning: Machine learning (ML) serves as the backbone of many modern malware detection systems. ML algorithms learn patterns from data, allowing for the classification and identification of malware based on its characteristics. Common techniques include decision trees, support vector machines (SVM), and random forests. These methods utilize labeled datasets to train models, which can then generalize to identify malware samples not seen during training. While effective, traditional ML techniques often struggle with high dimensional data and require extensive feature engineering [24]. Figure 1, depicts the architecture of machine learning lifecycle.

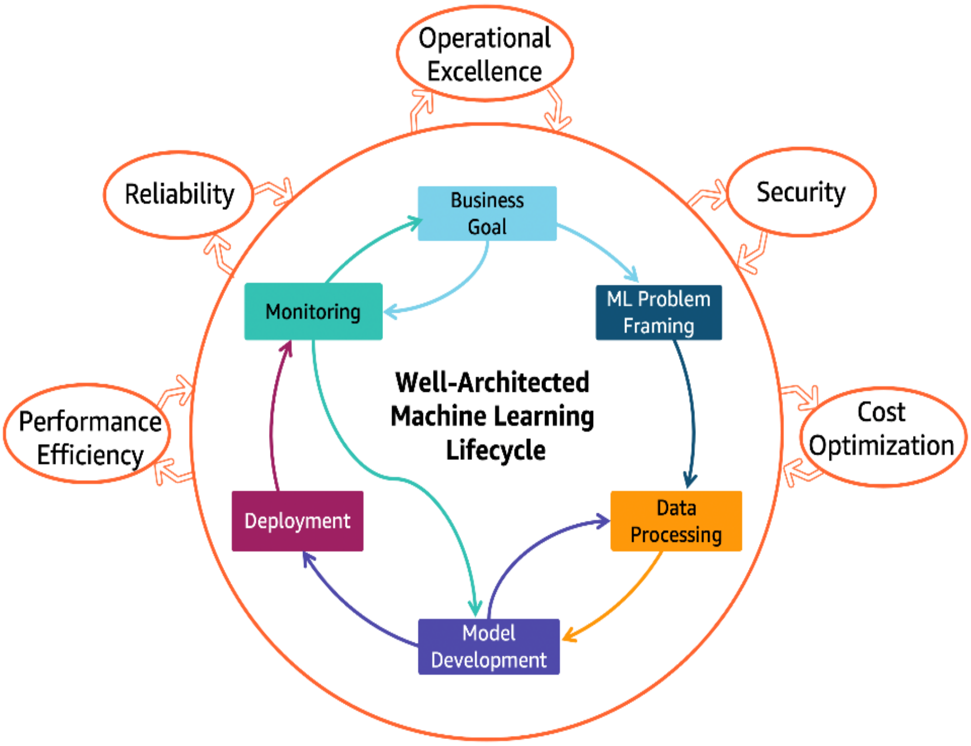

Figure 1: Architecture of Machine Learning Lifecycle [25].

2. Deep Learning: Deep learning (DL) is a branch of machine learning that utilizes neural networks with multiple layers to automatically extract features from raw data. This capability significantly improves detection rates, especially in high dimensional environments such as binary files or network traffic. Techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown promise in detecting malware through both static and dynamic analysis. The ability of deep learning models to handle large datasets and capture complex patterns has made them particularly valuable in the field of cybersecurity [26].

3. Hybrid Models: Hybrid models combine the strengths of both machine learning and deep learning approaches. By integrating various algorithms, these models aim to improve detection accuracy and robustness against a wide array of malware. Hybrid systems can use ML for initial feature extraction and DL for classification, thereby leveraging the advantages of both methodologies [27].

2.1 Advanced Deep Learning Techniques for Enhanced Specialized Applications

This section delves into advanced deep learning techniques that are specifically designed to boost performance in targeted application domains. By focusing on recent advancements, it highlights how specialized methods are employed to address unique challenges within these areas. The analysis emphasizes techniques that enhance model accuracy, computational efficiency, and scalability, offering a comprehensive view of their impact and potential for driving further innovation in focused deep learning applications.

1. Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) have become a valuable tool for the static analysis of malware binaries. By interpreting binary files as images or spectrograms, CNNs can detect patterns indicative of malicious behavior. The convolutional layers of these networks autonomously acquire spatial hierarchies of features, making them well-equipped to scrutinize the complex structures of binary code [28]. Figure 2, depicts Architecture of Convolutional Neural Networks (CNNs).

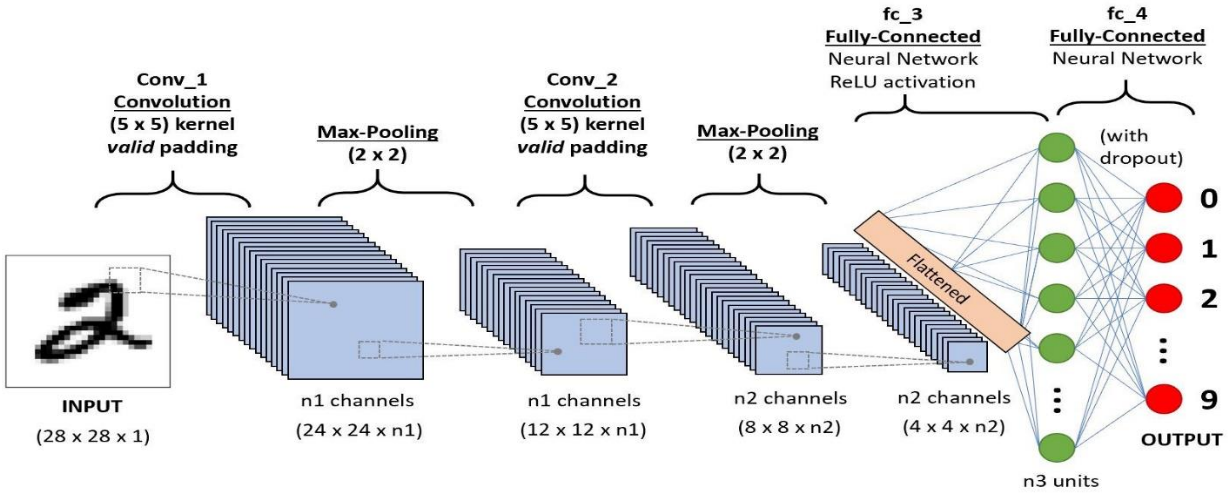

Figure 2: Architecture of Convolutional Neural Networks (CNNs) [29].

Applications in Static Analysis: CNNs are highly adept at detecting malware by analyzing the static characteristics of binaries, including opcode sequences and control flow graphs. Studies have demonstrated that CNNs can achieve exceptional accuracy in differentiating between malicious and benign binaries by extracting pertinent features without requiring extensive preprocessing. This capacity to directly learn from raw data simplifies the detection process, diminishing the need for manual feature extraction.

2. Recurrent Neural Networks (RNNs) and their advanced variant, Long Short Term Memory networks (LSTMs), are designed for dynamic analysis, where understanding sequential patterns is crucial. Malware often demonstrates behaviors that unfold over time, making sequence modeling essential for accurate detection [30]. Figure 3, illustrates the Architecture of Recurrent Neural Networks (RNNs).

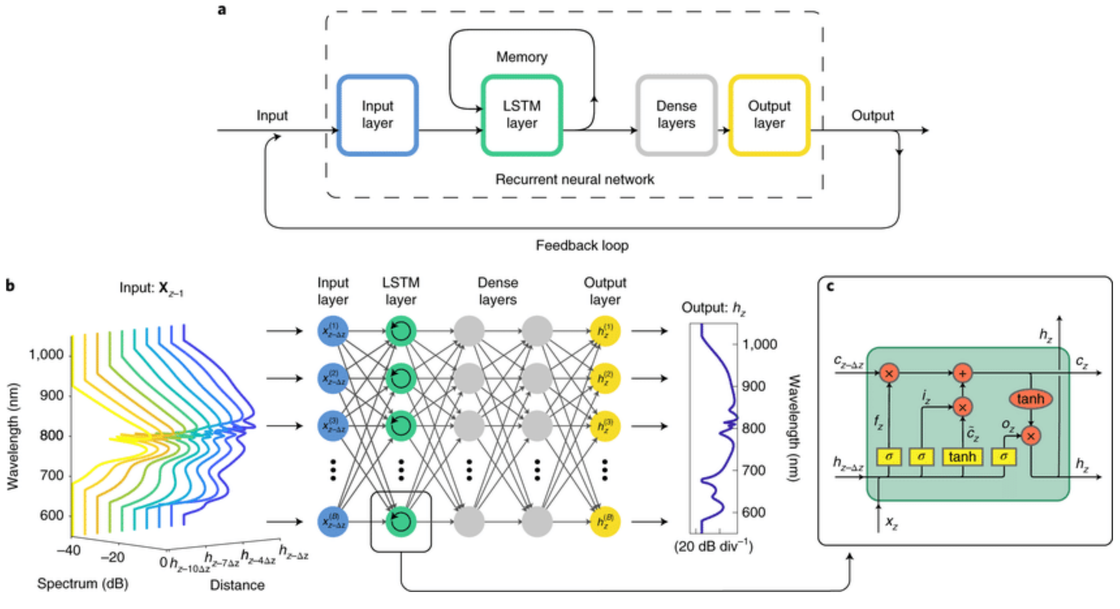

Figure 3: Architecture of Recurrent Neural Networks (RNNs) [31].

3. Applications in Dynamic Analysis: RNNs and LSTMs are capable of capturing the temporal dependencies of system calls and API interactions, enabling the detection of anomalous behavior patterns that may indicate the presence of malware. For example, a sequence of legitimate system calls could be disrupted by a series of suspicious calls, uncovering potential malicious activity. Through the analysis of these behavioral sequences, RNNs can accurately differentiate between benign and malicious processes, offering crucial real-time detection capabilities essential in dynamic environments [32].

4. Auto encoders and Anomaly Detection

Auto encoders are unsupervised neural networks designed to learn efficient representations of data through dimensionality reduction. They consist of an encoder that compresses input data into a lower dimensional representation and a decoder that reconstructs the input from this representation [33].

5. Use Cases in Detecting Novel or Unseen Threats: In malware detection, auto encoders can be employed to identify novel threats by analyzing deviations from learned normal behavior. By training an auto encoder on benign samples, the model can reconstruct inputs accurately. If a new sample yields a high reconstruction error, it may signify an anomaly, prompting further investigation. This approach is particularly beneficial for detecting previously unseen malware, as it does not rely on labeled data.

6. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) consist of two neural networks, the generator and the discriminator, that work in tandem. The generator creates synthetic data, while the discriminator evaluates its authenticity. This adversarial process leads to the generation of high quality data that mimics real world samples [34].

7. Applications in Simulating Malware Variants: In the context of malware detection, GANs can be utilized to simulate various malware variants, enhancing the training datasets used for detection models. By generating diverse malware samples, GANs enable more robust training of detection systems, ensuring they are better equipped to identify different strains of malware. Furthermore, GANs can also aid in improving the resilience of models against adversarial attacks, as they can generate adversarial examples for training, enhancing the overall robustness of detection systems [34].

2.2 Hybrid and Ensemble Methods

The integration of multiple AI models has gained traction in malware detection, leading to the development of hybrid and ensemble methods. These approaches capitalize on the strengths of various algorithms, enhancing overall detection performance.

i. Combination of Different AI Models

Hybrid models often combine machine learning and deep learning techniques to exploit their complementary strengths. For instance, a hybrid model might utilize traditional ML algorithms for initial feature extraction from network traffic data, followed by deep learning techniques for classification. This strategy not only improves accuracy but also reduces the computational burden associated with processing high dimensional data [35].

ii. Ensemble Methods: Ensemble methods further enhance detection rates by aggregating the predictions of multiple models. Techniques such as bagging and boosting are commonly employed to combine the outputs of different classifiers, leading to improved robustness and reduced over fitting. For example, Random Forests, an ensemble of decision trees, can achieve higher accuracy and stability compared to individual classifiers. By leveraging the collective intelligence of various models, ensemble methods can significantly enhance the performance of malware detection systems [36].

1. Bagging (Bootstrap Aggregating):

Bagging combines predictions from multiple models by averaging them, as shown in Equation (1). This approach improves model robustness by reducing variance in predictions [37].

$$\hat{y}_{\text{bagging}} = \frac{1}{M}\sum_{m=1}^{M} f_m(x) \quad \textit{Equation 1}$$

Where:

- $\hat{y}_{\text{bagging}}$ is the aggregated prediction,

- M is the total number of models in the ensemble,
- $f_m(x)$ represents the prediction of the mth model for input x.

2. Boosting:

Boosting uses weighted combinations of model predictions, with weights determined by model accuracy, as indicated in Equation (2). This enhances the model's ability to correct errors iteratively [38].

$$\hat{y}_{boosting} = \sum_{m=1}^{M} \alpha_m f_m(x) \quad \textit{Equation 2}$$

Where:

- $\hat{y}_{boosting}$ is the final boosted prediction,
- α_m is the weight for each model f_m, based on its accuracy,
- M is the total number of models.

3. Random Forest (Ensemble of Decision Trees):

Random Forests use an ensemble of decision trees to average predictions, improving stability and accuracy equation 3. Each tree contributes equally to the final prediction [39].

$$\hat{y}_{RF} = \frac{1}{N} \sum_{n=1}^{N} T_n(x) \quad \textit{Equation 3}$$

Where:

- $\hat{y}_{RF}$ is the final prediction from the Random Forest,
- N is the number of decision trees,
- $T_n(x)$ is the prediction from the nth tree for input x.

4. Accuracy of Ensemble Model:

The accuracy of an ensemble model, shown in Equation (4), benefits from combining multiple classifiers to reduce individual model errors, thus improving overall prediction reliability [39].

$$A_{ensemble} = 1 - \prod_{m=1}^{M} (1 - A_m) \quad \textit{Equation 4}$$

Where:

- $A_{ensemble}$ is the accuracy of the ensemble,
- A_m is the accuracy of individual model m,
- M is the number of models in the ensemble.

2.3 Artificial Intelligence driven Feature Extraction

Feature extraction plays a critical role in the effectiveness of malware detection systems. Automated methods for feature extraction can substantially improve the efficiency and accuracy of detection models.

i. Automated Feature Extraction Techniques:

AI driven approaches, particularly deep learning, have revolutionized the feature extraction process. Instead of relying on manual feature engineering, deep learning models can automatically learn relevant features from large scale datasets, such as binary code and network traffic. This capability not only streamlines the detection pipeline but also enhances the model's adaptability to evolving malware threats [40].

ii. Deep Learning for Binary Code Analysis: Deep learning techniques, particularly CNNs, have shown significant promise in analyzing binary code for feature extraction. By representing binaries as images, these networks can learn complex features directly from the data, improving the detection of malware without the need for extensive preprocessing. This automated feature extraction reduces the time and expertise required for manual feature engineering, making the detection process more efficient [41].

iii. Network Traffic Analysis: In the realm of network security, deep learning models can also analyze traffic patterns for feature extraction. By examining packet flows and communication sequences, these models can identify anomalies indicative of malware activity. Utilizing deep learning for network traffic analysis allows for the detection of subtle behavioral changes that may go unnoticed by traditional methods, enhancing the overall efficacy of malware detection systems [42].

Table 1. Provides the core techniques in AI-driven feature extraction and real-time malware detection.

Table 1. Core Techniques in AI-Driven Feature Extraction and Real-time Malware Detection

Ref No.	Technique	Application	Process	Advantages	Limitations
[40]	Automated Feature Extraction	Malware Detection.	• Leverages AI to extract key features from large datasets, such as binary code and network data.	• Increases detection efficiency and adaptability to new malware patterns.	• High computational demand and dependence on quality data.
[41]	Deep Learning for Binary Code Analysis	Binary Feature Extraction.	• Uses CNNs to analyze binaries as images, automatically learning complex features without manual input.	• Reduces manual feature engineering, enhances detection accuracy.	• Requires significant computational power; may need pre-training.
[42]	Network Traffic Analysis	Network Security.	• Examines packet flows and sequences to identify anomalous patterns linked to malware activity.	• Detects subtle behavioral anomalies, complements traditional methods.	• Risk of false positives due to network variability and needs continual updates.

3. Real Time Malware Detection Systems

Real-time malware detection systems are designed to identify and mitigate malicious activities as they occur, ensuring immediate protection for digital environments. These systems continuously monitor network traffic, system behavior, and data patterns to detect anomalies indicative of malware. Leveraging machine learning and deep learning algorithms, they analyze vast amounts of data to recognize potential threats, adapting to new forms of malware through adaptive learning mechanisms [43]. Additionally, real-time detection often incorporates signature based and behavior based approaches to enhance accuracy and response speed. By identifying threats as they emerge, these systems help minimize damage and maintain system integrity, making them essential in today's cybersecurity landscape. Real-time systems are integral to protecting critical infrastructures where latency and prompt threat mitigation are crucial.

3.1 Challenges in Real Time Detection

As cyber threats grow more sophisticated, the demand for real time malware detection systems has become increasingly critical. These systems aim to identify and neutralize threats as they emerge, minimizing potential damage. However, deploying AI models in real time environments presents several key challenges that must be addressed to ensure effectiveness.

1. Speed and Latency Concerns

One of the most significant challenges in real time malware detection is achieving the necessary speed for timely responses. Cyber-attacks can occur within milliseconds, and any delay in detection can allow malware to execute its malicious payload, leading to severe consequences. Therefore, AI models must be optimized to process data quickly and deliver results in real time. Traditional deep learning models, while effective, often require substantial computational resources and time for inference. This latency can be detrimental in real time environments where immediate action is essential. As such, balancing the complexity of AI models with the speed of detection becomes a critical consideration [44].

2. Accuracy Tradeoffs

While speed is paramount, the accuracy of malware detection is equally vital. There is an inherent tradeoff between speed and accuracy, where optimizing for one can compromise the other. For example, simplified models designed for faster inference may not capture the nuanced patterns necessary for high accuracy detection. Conversely, more complex models that offer improved accuracy often result in longer processing times. Maintaining a low false positive rate is particularly challenging in real time detection. A high rate of false positives can lead to unnecessary alerts, overwhelming security teams and resulting in alert fatigue. Thus, achieving a balance between speed, accuracy, and false positive rates is essential for the effective deployment of AI in real time malware detection [44].

3. Resource Constraints

Real time detection systems must also operate under various resource constraints, particularly in environments such as edge computing or Internet of Things (IoT) devices. These environments often have limited computational power, memory, and bandwidth, making it challenging to deploy large-scale AI models. Security solutions must, therefore, be designed to be lightweight and efficient to function effectively within these constraints [45].

3.2 Case Studies and Applications

The implementation of real-time AI systems for malware detection has been observed in both industry and academia, showcasing their effectiveness in real-world scenarios.

1. Microsoft Defender

Microsoft Defender employs advanced machine learning and deep learning techniques to provide real-time protection against malware. Utilizing behavioral analysis, Microsoft Defender detects threats based on patterns of malicious activity rather than relying solely on signature-based detection. The system continually updates its models using data from millions of devices worldwide, allowing it to adapt quickly to emerging threats. One of the key features of Microsoft Defender is its use of cloud based AI models that leverage vast computational resources. This hybrid approach enables quick processing of large datasets while providing real-time protection on individual devices. Furthermore, its integration with threat intelligence feeds enhances its ability to swiftly detect and respond to known vulnerabilities and emerging threats [46].

2. VirusTotal

VirusTotal, a widely used malware analysis tool, incorporates AI-driven techniques for real-time detection. It aggregates results from multiple antivirus engines and utilizes machine learning models to analyze files and URLs for potential threats. The platform allows users to submit samples for analysis and returns results within seconds.

VirusTotal employs ensemble methods that combine the outputs of various detection engines, improving overall accuracy and reducing false positives. Its real time analysis capabilities have made it a valuable resource for security researchers and organizations seeking to identify and mitigate threats promptly [47].

3. Academic Initiatives

In academia, various research initiatives have focused on developing real time malware detection systems using AI. For example, research groups have explored the use of deep learning models to analyze system call sequences in real time, detecting anomalies indicative of malware behavior. These systems can operate with minimal latency, providing immediate feedback on potential threats. Other academic studies have investigated the application of reinforcement learning for adaptive malware detection. By continuously learning from the environment, these systems can adjust their detection strategies in real time, improving their resilience against evolving threats.

3.3 Evaluating Model Performance in Real Time

Evaluating the performance of real time malware detection systems requires specific metrics and benchmarks that reflect their operational effectiveness. Key performance indicators include latency, false positive rates, and accuracy.

1. Latency

Latency is a critical metric for real time detection systems, as it measures the time taken from data input to the generation of a detection output. In a malware detection context, low latency is essential to ensure that threats are identified and mitigated before they can inflict damage. Evaluating latency involves assessing the time required for data preprocessing, model inference, and any post processing required to interpret the results [48].

2. False Positive Rates

The false positive rate (FPR) is another crucial metric, representing the percentage of benign instances incorrectly classified as malicious. High false positive rates can overwhelm security teams with alerts, leading to alert fatigue and potentially causing real threats to be overlooked. Evaluating the FPR involves analyzing the model's predictions against a labeled dataset, allowing researchers to gauge its reliability and effectiveness in distinguishing between malicious and benign activity [49].

3. Accuracy

Accuracy is a fundamental measure of a detection system's performance, indicating the proportion of correct predictions made by the model. However, it is essential to evaluate accuracy in conjunction with other metrics, such as precision and recall, to obtain a comprehensive understanding of a model's performance. Precision measures the proportion of true positives among all positive predictions, while recall (or sensitivity) assesses the model's ability to identify all actual positive cases [50].

4. Benchmarking

To provide a standardized basis for evaluating real time malware detection systems, benchmarking datasets and frameworks are necessary. Public datasets, such as the Microsoft Malware Classification Challenge and the Malware Training Set from the Kaggle platform, offer a wealth of labeled samples for training and evaluating models. These datasets facilitate comparisons between different detection approaches, ensuring that advancements in real time malware detection can be effectively measured and communicated within the research community [51].

Table 2, illustrates the evaluation metrics for real-time and adversarially resilient malware detection models

Table 2. Evaluation Metrics for Real Time and Adversarially Resilient Malware Detection Models

Ref No.	Evaluation Metric	Description	Measurement	Benchmark Datasets	Improvement Techniques
[48]	**Latency**	Time from data input to detection response	Average Latency (ms)	Real-time simulation datasets, proprietary test environments	Model optimization, streamlined data preprocessing

[49]	**False Positive Rate (FPR)**	Rate of benign samples incorrectly flagged as malicious	FPR (%)	Kaggle Malware Training Set, Microsoft Malware Classification Challenge	Feature refinement, adaptive threshold tuning
[50]	**Accuracy**	Proportion of correct classifications in total predictions	Accuracy (%), Precision, Recall	Mixed datasets (static/dynamic), Microsoft Malware Classification, custom labeled datasets	Balancing recall/precision, enhancing training data quality
[51]	**Benchmarking**	Standardized datasets for consistent model evaluation	Performance Index	Microsoft Malware Classification, Kaggle datasets, real-time organizational data	Cross dataset testing, integration with real-world data

4. Adversarial Resilience in Malware Detection

Adversarial resilience in malware detection focuses on strengthening detection models to withstand manipulation attempts by attackers who try to evade detection. As malware creators increasingly employ adversarial tactics, such as modifying malware signatures or altering code structure, robust detection models become essential. Techniques like adversarial training, model hardening, and the use of generative adversarial networks (GANs) help improve resilience by making models more adaptable to adversarially altered inputs. Additionally, enhancing model interpretability and incorporating anomaly detection further fortifies systems against these evolving threats. Building resilience against adversarial tactics is crucial to maintaining the reliability and robustness of AI driven malware detection frameworks.

4.1 Prolog of Adversarial Machine Learning

Adversarial machine learning has emerged as a critical area of concern within the field of cyber security, particularly in malware detection systems. As AI and machine learning models become increasingly prevalent in identifying malicious software, adversaries have devised sophisticated methods to circumvent these defenses. Understanding the dynamics of adversarial attacks is crucial for developing robust malware detection systems capable of withstanding such threats. Adversarial attacks are malicious attempts to subtly manipulate input data to deceive AI models into making incorrect predictions. They can be broadly categorized into two main types: evasion attacks and poisoning attacks. Figure 4. Depicts diverse types of adversarial attacks and their classification [39].

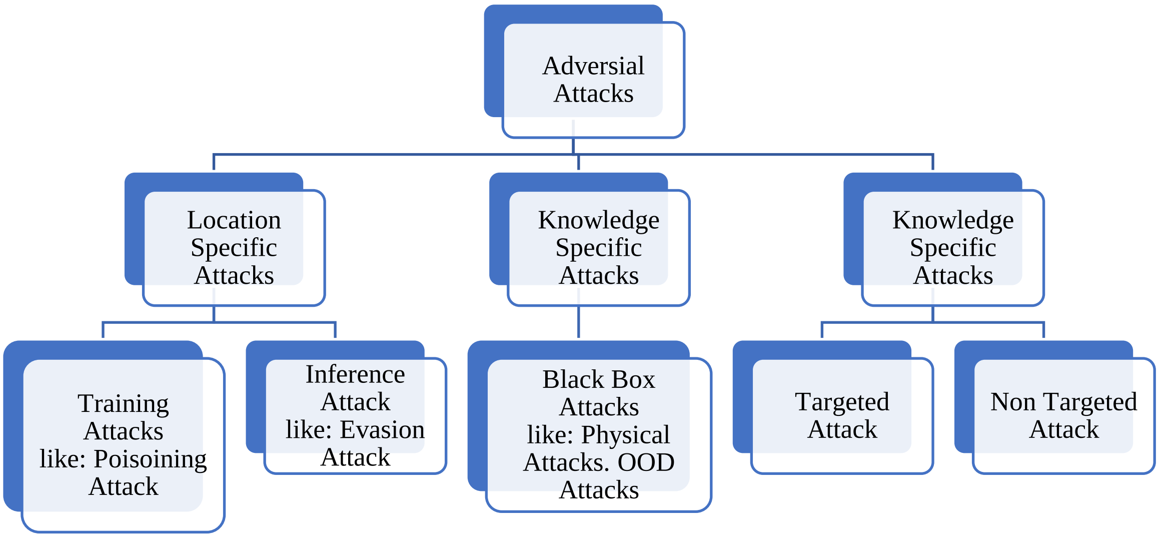
Adversial Attacks
Location Specific Attacks
Knowledge Specific Attacks
Knowledge Specific Attacks
Training Attacks like: Poisoining Attack
Inference Attack like: Evasion Attack
Black Box Attacks like: Physical Attacks. OOD Attacks
Targeted Attack
Non Targeted Attack

Figure 4. Diverse Types of Adversarial Attacks and their Classification [39].

1. Evasion Attacks

Evasion attacks occur when an attacker manipulates input data at the time of inference to evade detection. In the context of malware detection, this could involve altering a piece of malicious software so that it appears benign to the detection system. For instance, an adversary might modify the binary code or introduce noise to the input features, effectively disguising the malware's true nature. The success of such attacks poses a significant threat to the reliability of AI driven malware detection systems, as they can lead to false negatives, allowing malware to execute without detection.

2. Poisoning Attacks

Poisoning attacks, on the other hand, involve compromising the training process of a machine learning model by injecting malicious samples into the training dataset. An adversary can manipulate the model to learn incorrect patterns or biases by strategically introducing poisoned data, ultimately degrading its performance. In malware detection, this can result in a model that is less capable of recognizing genuine malware, thus rendering the detection system vulnerable to future attacks.

3. Impact on Malware Detection Systems

The impact of adversarial attacks on malware detection systems is profound. They can undermine the very foundation of trust in AI driven security solutions, leading to increased risks of undetected malware. As adversarial techniques evolve, it becomes essential for researchers and practitioners to develop effective defensive measures to ensure the resilience of malware detection systems.

4.2 Defensive Techniques

In response to the growing threat of adversarial attacks, several defensive techniques have been proposed to enhance the robustness of malware detection systems. These techniques aim to improve model resilience by minimizing vulnerabilities to adversarial manipulations.

1. Adversarial Training

Adversarial training is a prominent strategy that involves augmenting the training dataset with adversarial examples. By exposing the model to these adversarially perturbed inputs during the training process, the model learns to identify and correctly classify both benign and malicious samples, including those designed to evade detection.

The process typically involves the following steps:

1. Generate Adversarial Examples: Using techniques such as the Fast Gradient Sign Method (FGSM) or Projected Gradient Descent (PGD), adversarial examples are generated based on the current model’s parameters.

2. Augment Training Data: The generated adversarial examples and legitimate samples are incorporated into the training dataset.

3. Retrain the Model: The model is retrained using this augmented dataset, allowing it to learn to recognize adversarial examples.

Adversarial training has been shown to enhance model robustness, but it also requires careful consideration of the tradeoffs involved, including the potential for increased training time and computational costs.

2. Feature Squeezing and Defensive Distillation

Feature squeezing is a defensive technique designed to reduce a model's vulnerability to small perturbations in input data. This approach involves compressing the input space by removing certain features or applying transformations that simplify the input representation. For example, quantization can be used to reduce the precision of input data, making it more difficult for adversaries to find effective perturbations that mislead the model.

Defensive distillation is another technique that focuses on training a model to produce softened output probabilities, which can improve robustness against adversarial attacks. This process involves training a secondary model on the outputs of a primary model, allowing the secondary model to learn a smoother decision boundary. By distilling knowledge from the first model, the second model becomes less sensitive to small input changes, enhancing its resilience to adversarial perturbations.

3. Various Defense Mechanisms

In addition to adversarial training and feature squeezing, several other defense mechanisms have been proposed to bolster the resilience of malware detection systems against adversarial attacks:

a. Gradient Masking: This technique aims to obscure the gradients of the model, making it more challenging for adversaries to compute effective perturbations. By altering the loss landscape, gradient masking can slow down or hinder the effectiveness of adversarial attacks. However, it is worth noting that gradient masking may not provide robust protection against determined adversaries, as they can often find alternative attack vectors.

b. Input Transformation: Input transformation involves preprocessing input data to mitigate the effects of adversarial perturbations. Techniques such as image filtering, JPEG compression, or adding noise can obscure the impact of adversarial modifications, improving the model's ability to maintain accuracy.

c. Anomaly Detection: Incorporating anomaly detection mechanisms into malware detection systems can provide an additional layer of defense. By identifying unusual patterns in input data that deviate from expected behavior, these systems can flag potential adversarial inputs and prevent them from influencing the model's predictions.

4.3 Comparative Analysis

A comparative analysis of different defensive techniques highlights the strengths and weaknesses of each approach in enhancing adversarial resilience in malware detection systems.

A. Adversarial Training vs. Feature Squeezing

Adversarial Training and Feature Squeezing are two prominent techniques for enhancing model robustness against adversarial attacks. Adversarial Training involves augmenting the training dataset with adversarially modified examples, allowing the model to learn and adapt to potential attack patterns. This method enhances the model’s ability to recognize and resist adversarial inputs but can be computationally intensive. In contrast, Feature Squeezing reduces the input feature space by applying transformations, such as bit depth reduction or smoothing, to limit the exploitable variance in data. While feature squeezing is computationally lighter, it may not be as comprehensive as adversarial training. Both techniques contribute to improved adversarial resilience but offer distinct tradeoffs in terms of complexity, performance, and robustness [52]. Table 3. Illustrated the strengths and weakness of adversarial training and feature squeezing.

Table 3. Strengths and Weakness of Adversarial Training and Feature Squeezing.

Ref No.	Technique	Strengths	Weakness
[52]	Adversarial Training	• Enhances robustness against a wide range of adversarial examples. • Allows the model to learn directly from adversarial perturbations.	• Increases training time and computational resource requirements. • May lead to over fitting if not carefully managed.
[52]	Feature Squeezing	• Simplifies input space, making it harder for adversaries to find effective perturbations. • Can be easily integrated into existing detection systems.	• May result in a loss of critical information, potentially affecting accuracy. • Effectiveness depends on the choice of features to squeeze.

B. Defensive Distillation and Gradient Masking

Defensive Distillation and Gradient Masking are two key strategies for improving model robustness against adversarial attacks. Defensive Distillation works by training models on softened outputs (or probabilities) rather than hard class labels, which creates smoother decision boundaries and makes it more difficult for minor input perturbations to mislead the model. This approach also benefits generalization by forcing the model to focus on broader patterns. However, it comes with a tradeoff in computational cost due to the additional training and may lose effectiveness against adaptive attacks designed to bypass this defense. On the other hand, Gradient Masking aims to obscure a model's gradients, reducing the attackers' ability to craft perturbations by hiding the model’s sensitivity to input changes. This method is relatively simple to implement and can quickly add a layer of protection. However, gradient masking has been criticized for offering only a temporary shield, as determined attackers can still bypass it by exploiting weaknesses in the technique. Additionally, it may not defend effectively against all adversarial attacks, as certain advanced methods can still find ways around the obscured gradients [53]. Table 4. Illustrates the strengths and weakness of defensive distillation and gradient masking

Table 4. Strengths and Weakness of Defensive Distillation and Gradient Masking

Ref No.	Technique	Strengths	Weaknesses
[53]	**Defensive Distillation**	• Provides a smoother decision boundary, enhancing resilience to small perturbations. • Can improve model generalization by focusing on softened outputs.	• Requires additional training, increasing computational costs. • Effectiveness may diminish against adaptive adversarial attacks.
[53]	**Gradient Masking**	• Can obscure the model's sensitivity to perturbations, complicating the adversarial attack process. • Provides a straightforward implementation for model hardening.	• Often criticized for offering a false sense of security; determined adversaries can circumvent it. • May not effectively protect against all forms of adversarial attacks.

C. Anomaly Detection and Input Transformation

Anomaly Detection and Input Transformation are two widely used techniques to strengthen adversarial resilience in machine learning systems. Anomaly Detection works by identifying unusual or suspicious input patterns, acting as an additional layer of defense against potential threats. This approach enhances the robustness of existing detection systems by flagging inputs that deviate from normal behavior, potentially signaling an attack. However, it requires precise tuning to distinguish between legitimate variations and truly malicious activity, as misconfigurations can lead to false positives, causing unnecessary alerts that may overwhelm security teams. Input Transformation, on the other hand, involves preprocessing inputs to mitigate adversarial perturbations before they reach the model. This method can be seamlessly integrated into existing systems, reducing the impact of adversarial inputs without the need for model retraining. Nevertheless, input transformations may inadvertently alter benign data, potentially affecting the accuracy of legitimate inputs, and the choice of transformation greatly influences its effectiveness against specific adversarial tactics. Both techniques offer valuable contributions to system security but require careful implementation to optimize effectiveness and minimize drawbacks [54]. Table 5. Illustrates the strengths and weakness of anomaly detection and input transformation

Table 5. Strengths and Weakness of Anomaly Detection and Input Transformation

Ref No.	Technique	Strengths	Weaknesses
[54]	Anomaly Detection	• Adds an additional layer of defense by identifying suspicious input patterns. • Can complement existing detection systems, enhancing overall security	• May produce false positives, leading to unnecessary alerts. • Requires careful tuning to distinguish between legitimate and malicious behavior

[54]	Input Transformation	• Provides a practical approach to mitigating the effects of adversarial perturbations. • Can be applied as a preprocessing step, making it easy to integrate into existing systems.	• May inadvertently modify benign inputs, potentially affecting detection accuracy. • The effectiveness depends on the specific transformation applied

The effectiveness depends on the specific transformation applied.

4.4 Open Challenges in Adversarial Resilience

Despite the advancements in adversarial resilience techniques, several ongoing research challenges remain. Addressing these challenges will be crucial for developing more robust malware detection systems that can withstand increasingly sophisticated adversarial attacks [55].

1. Balancing Accuracy and Robustness

One of the primary challenges in developing adversarially resilient models is striking the right balance between accuracy and robustness. While defensive techniques can enhance a model's ability to withstand adversarial attacks, they may also lead to tradeoffs in accuracy. Future research must focus on refining defensive strategies that maintain high accuracy while improving robustness against adversarial manipulations.

2. Adaptive Attacks and Defenses

As adversarial techniques evolve, so too must the defensive mechanisms employed to counteract them. The development of adaptive attacks—where adversaries modify their attack strategies based on the defensive measures in place—poses a significant challenge. Ongoing research should explore dynamic defense mechanisms that can adapt to emerging threats, ensuring that malware detection systems remain effective in the face of evolving adversarial tactics.

3. Real World Application and Generalization

Another critical challenge lies in the real world application of adversarial resilience techniques. Many existing studies focus on controlled environments with specific datasets, which may not accurately reflect the complexity of real world scenarios. Research efforts should emphasize the development and testing of adversarial defenses in more diverse, real world contexts to ensure their effectiveness across various applications.

4. Continuous Learning and Updating Models

Incorporating continuous learning mechanisms into malware detection systems can enhance their ability to adapt to new threats and adversarial techniques. Developing systems that can automatically update and retrain on new data while maintaining robustness is an ongoing challenge. Future research should explore the integration of online learning and continual adaptation strategies to improve the resilience of malware detection systems against adversarial attacks. Table 6. Depicts the open challenges in adversarial resilience with solutions.

Table 6. Open Challenges in Adversarial Resilience with solutions

Ref No.	**Open Challenges**	**Limitations**	**Examples**	**Solutions**
[55]	**Balancing Accuracy and Robustness**	Defensive techniques can enhance resilience but often lead to reduced accuracy,	A malware detection model with adversarial training may identify threats	Develop hybrid models that adjust the degree of adversarial training

		making it difficult to maintain both high accuracy and robustness in adversarial settings.	but also misclassify benign inputs.	based on data sensitivity, or use ensemble models to balance tradeoffs.
[55]	**Adaptive Attacks and Defenses**	Attackers continuously evolve their strategies, making static defenses vulnerable to adaptive attacks. Adversaries can modify their methods to bypass current defenses, leaving systems exposed.	Attackers adjusting their methods to evade an AI based detection model by manipulating patterns of malware signatures.	Design dynamic defense mechanisms, such as metal earning approaches, which allow the model to detect and adapt to new patterns over time.
[55]	**Real World Application and Generalization**	Adversarial resilience methods often focus on limited, controlled datasets, reducing their applicability in diverse, real-world environments where data is more variable	A model tested in lab settings may fail in real-world scenarios with unanticipated malware variations.	Conduct tests in more diverse environments and incorporate domain adaptation techniques to improve generalization across varied datasets.
[55]	**Continuous Learning and Updating Models**	Maintaining robustness in models that continually learn from new data without manual retraining is challenging. Continuous updates risk introducing noise, potentially diminishing model reliability over time.	A real-time malware detection system may lose accuracy if newly added data is adversarial or noisy, impacting its detection capabilities.	Implement online learning techniques with robust regularization and use periodic validation checks to ensure the integrity of updates over time.

5. Comparative Analysis of Techniques

This section evaluates various methods used in malware detection, highlighting each approach's strengths and limitations. By examining core metrics such as accuracy, latency, robustness, and adaptability, this analysis provides insights into how different models perform across real-world and controlled environments. Techniques like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and ensemble methods are compared to reveal their effectiveness in specific scenarios, such as handling adversarial attacks or maintaining low false positive rates. This comparison allows researchers to understand the tradeoffs between precision, recall, and computational demands, ultimately guiding the selection of optimal models for different malware detection contexts.

5.1 Performance Metrics

When assessing the effectiveness of various AI models for malware detection, performance metrics play a crucial role in providing a comprehensive view of their capabilities. Commonly used metrics include accuracy, precision, recall, and F1 score, each offering unique insights into a model's performance across different malware detection tasks [7].

In classification metrics, **True Positives (TP)** are cases correctly identified as positive (e.g., malware correctly flagged as malware), while **True Negatives (TN)** are instances correctly identified as negative (e.g., benign files correctly flagged as benign). **False Positives (FP)** occur when benign instances are mistakenly classified as threats, and **False Negatives (FN)** happen when true threats are misclassified as benign.

a. Accuracy

Accuracy measures the overall proportion of correctly classified instances among the total number of instances showed in equation 5. While it provides a general sense of model performance, accuracy can be misleading in cases of class imbalance, such as in malware detection, where benign instances often far outnumber malicious ones. For example, if a model correctly identifies 95% of benign samples but fails to detect 20% of malware, the high accuracy may give a false sense of security. Equation 1. Depicts how to calculate the Accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \textit{Equation 5}$$

b. Precision

Precision indicates the proportion of true positive predictions (correctly identified malware) among all positive predictions made by the model illustrated in equation 6. This metric is particularly important in malware detection, as a high precision value indicates that the system generates fewer false positives, reducing unnecessary alerts. In scenarios where security teams may be overwhelmed by alerts, high precision can lead to more effective incident response. Equation 2. Depicts how to calculate the Precision.

$$\text{Precision} = \frac{TP}{TP+FP} \quad \textit{Equation 6}$$

c. Recall

Recall, also known as sensitivity, measures the proportion of true positive predictions among all actual positive cases (all malware present in the dataset) as shown in equation 7. A high recall value signifies that the model is effective at identifying malware, reducing the risk of false negatives. In malware detection, a focus on recall is critical, as failing to detect malware can lead to severe consequences, including data breaches or system compromises. Equation 3. Depicts how to calculate the Recall.

$$\text{Recall} = \frac{TP}{TP+FN} \quad \textit{Equation 7}$$

d. F1 Score

The F1 score is the harmonic mean of precision and recall, providing a balanced assessment of a model's performance provided in equation 8. It is particularly useful in scenarios where there is a tradeoff between precision and recall. A high F1 score indicates that the model maintains a balance between minimizing false positives and maximizing true positive detections. This metric is especially valuable in malware detection tasks where both precision and recall are important for operational effectiveness. Equation 4. Depicts how to calculate the F1 Score.

$$\text{F1 Score} = 2\,\frac{Precision * Recall}{Precision + Recall} \quad \textit{Equation 8}$$

These metrics provide a concise yet comprehensive assessment of model performance, balancing detection accuracy with error minimization.

5.2 Comparative Results

In recent studies, various AI models have been evaluated using these metrics across different malware detection tasks [56].

Convolutional Neural Networks (CNNs): Typically achieve high accuracy (over 95%) and excellent precision and recall rates (above 90%) when applied to static analysis of malware binaries. Their performance, however, can vary based on the quality of training data and the presence of adversarial examples.

Recurrent Neural Networks (RNNs): Often show high recall rates due to their ability to capture sequential dependencies in dynamic analysis. However, precision may vary, particularly if the training data includes noisy or adversarial samples.

Ensemble Models: These models tend to excel in achieving high F1 scores as they leverage the strengths of multiple algorithms. They can maintain a balance between precision and recall, making them particularly effective in complex detection tasks.

Table 7. Illustrates the performance comparison of CNNS, RNNS, and ensemble models in malware detection across different datasets and metrics.

Table 7. Performance Comparison of CNNs, RNNs, and Ensemble Models in Malware Detection across Different Datasets and Metrics

Ref No.	Model	Accuracy	Datasets	Key Findings	Performance Metrics
[56]	**Convolutional Neural Networks (CNNs)**	High accuracy, typically over 95%	Static malware binaries dataset	Achieves precision and recall rates above 90% when applied to clean datasets, but accuracy can drop in presence of adversarial examples.	Precision: >90%, Recall: >90%, Accuracy: ~95%+
[56]	**Recurrent Neural Networks (RNNs)**	Variable accuracy, moderate to high recall	Dynamic malware behavior datasets	High recall rates due to sequential pattern learning, but precision decreases with noisy or adversarial data.	Recall: High (often >90%), Precision: Moderate (~8090%)
[56]	**Ensemble Models**	Consistently high, typically above 92%	Mixed (static and dynamic) malware datasets	Excels in balancing precision and recall, achieving high F1 scores in complex detection tasks across diverse datasets.	F1 Score: High (90%+), Precision and Recall: Balanced (90%+)

The ideal model choice depends on the malware detection task, as each offers distinct strengths and weaknesses across performance metrics.

5.3 Tradeoffs in Real Time Systems

In real time malware detection systems, achieving the right balance between detection speed and accuracy is a critical consideration. Different models and approaches may exhibit various tradeoffs, impacting their practicality and effectiveness in real time environments [49][57].

1. Detection Speed

Detection speed is essential in real time malware detection, as even small delays can allow malware to execute its payload, leading to significant damage. Lightweight models, such as those based on decision trees or shallow neural networks, generally provide faster inference times compared to deeper architectures like CNNs or RNNs. However, faster models may sacrifice accuracy, potentially resulting in higher false negative rates.

2. Accuracy

On the other hand, more complex models, such as deep learning architectures, often deliver higher accuracy rates but at the cost of increased latency. For example, CNNs, while providing excellent performance in detecting known malware, may require longer processing times due to their depth and computational complexity. This tradeoff becomes particularly pronounced in scenarios where rapid decision making is crucial, such as in endpoint protection or intrusion detection systems.

3. Hybrid Approaches

To address these tradeoffs, hybrid approaches that combine lightweight models for initial screening and deeper models for more comprehensive analysis are increasingly being adopted. This allows for faster detection of obvious threats while still providing the capability to analyze more complex samples. Table 8 outlines key performance factors influencing real-time malware detection systems, highlighting tradeoffs and mitigation approaches to enhance both detection speed and accuracy.

Table 8: Tradeoffs in Real Time Malware Detection Systems

Ref No.	Performance Factor	Description	Example Models	Tradeoffs	Approach to Mitigate Tradeoffs
[57]	**Detection Latency**	Time taken to detect malware presence, critical to prevent immediate threats.	Decision Trees, Shallow Neural Networks	Fast inference but lower accuracy, risk of higher false negatives	Use lightweight models for initial screening in hybrid architectures
[57]	**Detection Accuracy**	Proportion of correct malware detections, essential for ensuring reliable identification of threats.	CNNs, RNNs	High accuracy but increased latency due to computational demands	Combine high accuracy models with faster models for selective deeper analysis
[57]	**Hybrid Model Approaches**	Integration of lightweight and deep models to balance detection speed and precision.	Lightweight models + CNN/RNN	Enables rapid initial detection, followed by comprehensive analysis	Implement tiered detection: quick initial assessment followed by in-depth validation for flagged samples

5.4 Adversarial Robustness Comparison

As adversarial attacks become more sophisticated, evaluating the robustness of different AI models against such threats is vital for ensuring the reliability of malware detection systems. Key studies have compared the resilience of various models, shedding light on their strengths and weaknesses in the face of adversarial manipulations [58].

a. CNNs

Convolutional Neural Networks have shown considerable robustness against adversarial attacks when appropriately trained with adversarial examples through adversarial training techniques. Studies have reported that CNNs can maintain a reasonable accuracy drop (around 510%) when faced with standard adversarial attacks. However, they can still be vulnerable to more sophisticated attacks that exploit their inherent weaknesses.

b. RNNs

Recurrent Neural Networks, particularly Long Short Term Memory networks (LSTMs), have been found to exhibit a unique vulnerability to adversarial perturbations in sequential data. Research indicates that while RNNs are effective at capturing temporal dependencies, their reliance on sequential input makes them susceptible to crafted sequences designed to confuse the model. As a result, adversarial attacks can lead to significant performance degradation (up to 30% drop in recall) for RNN based systems.

c. Ensemble Models

Ensemble models tend to demonstrate enhanced robustness against adversarial attacks. By aggregating the predictions of multiple models, they can effectively reduce the impact of adversarial examples that may deceive individual classifiers. Studies suggest that ensemble techniques can achieve up to 20% better resilience against adversarial attacks compared to single models, making them an appealing choice for enhancing adversarial robustness in malware detection systems.

Table 9. Comparative Analysis of Adversarial Robustness across CNNs, RNNs, and Ensemble Models in Malware Detection

Ref No.	Model	Adversarial Robustness	Observed Performance Drop	Strengths	Weaknesses
[58]	**CNNs**	Moderate to High	510% accuracy drop with standard adversarial attacks	• Robust with adversarial training, can maintain performance with typical adversarial examples	• Vulnerable to complex, highly crafted attacks that exploit CNN structure
[58]	**RNNs (e.g., LSTMs)**	Moderate	Up to 30% recall drop in presence of crafted sequences	• Effective in sequential pattern recognition, captures temporal dependencies	• High vulnerability to adversarial sequences, leading to significant recall loss under attack
	Ensemble Models	High	Up to 20% greater resilience compared to individual models	• Aggregates multiple model predictions, enhancing defense by reducing impact of adversarial examples	• Increased complexity and resource use; may still be partially susceptible to targeted ensemble attacks

Table 9. Provides a structured comparison of CNNs, RNNs, and ensemble models in terms of their resilience to adversarial attacks, highlighting the observed performance impacts and specific strengths and weaknesses relevant to malware detection tasks.

5.5 Open Challenges in Malware Detection

As the landscape of cyber security continues to evolve, the integration of Artificial Intelligence (AI) in malware detection presents several unresolved issues. While significant advancements have been made, key challenges remain, particularly concerning adversarial resilience and real time processing capabilities shown in table 10 [59].

i. Adversarial Resilience

One of the most pressing challenges in AI driven malware detection is enhancing adversarial resilience. Despite the development of various defensive techniques, such as adversarial training and input transformation, these methods often fall short against sophisticated adversarial attacks. The arms race between adversaries and defenders means that as detection systems improve, attackers will continually adapt their strategies to exploit vulnerabilities. Moreover, many existing defenses lack robustness against adaptive adversarial techniques, where attackers modify their approaches based on the defensive mechanisms deployed. Research must focus on developing more dynamic and robust defensive strategies that can evolve in tandem with adversarial methods. This entails not only creating resilient models but also understanding the broader landscape of adversarial behaviors and techniques.

ii. Real Time Processing

The need for real time processing in malware detection systems poses another significant challenge. Detecting and responding to malware threats in real time is critical for mitigating damage and preventing breaches. However, achieving the necessary speed without compromising accuracy is a delicate balance that many models struggle to maintain. In real world environments, especially those involving edge computing or IoT devices, resource constraints further complicate the deployment of high performance models. Lightweight models often sacrifice accuracy for speed, while more complex architectures, such as deep learning models, may introduce unacceptable latency. Research efforts should focus on optimizing existing models to improve their speed without losing the robustness required for effective malware detection.

iii. Model Interpretability

Model interpretability is a critical issue that remains largely unresolved in AI driven malware detection. Many advanced AI models, especially deep learning architectures, function as “black boxes,” making it challenging for security analysts to understand their decision making processes. This lack of transparency can hinder trust in AI systems, especially in high stakes environments where decisions can have significant repercussions. Improving model interpretability is essential for fostering user trust and enabling effective human AI collaboration in cyber security. Security analysts need to understand how and why certain decisions are made, mainly when a model flags a benign application as malicious or vice versa. Future research should develop methods that enhance interpretability without sacrificing performance, such as explainable AI techniques that provide insights into the reasoning behind model predictions.

iv. Scalability and Adaptability

As malware evolves, so too must the models designed to detect it. Scalability and adaptability are crucial attributes that many current models lack. Traditional training methods often rely on static datasets that do not adequately reflect the dynamic nature of malware threats. Consequently, models can become outdated quickly, rendering them ineffective against new malware strains. To address this challenge, ongoing research should explore methodologies that allow models to adapt to emerging threats continually. This may involve developing self-learning systems that can update their parameters based on new data or integrating feedback loops that incorporate real-time threat intelligence. Additionally, enhancing the scalability of models to handle large-scale data inputs while maintaining performance will be vital for effective deployment in the real world.

Table 10, provides the examination of key open challenges and future directions in AI powered malware detection.

Table 10: Key Open Challenges and Future Directions in AI Driven Malware Detection

Ref No.	Challenge	Description	Current Limitations	Future Directions
[59]	Adversarial Resilience	• Enhancing the model's ability to withstand adversarial attacks through robust defensive strategies.	• Existing defenses like adversarial training and input transformation often fail against adaptive adversarial techniques.	• Develop dynamic, adaptable defense strategies that evolve with adversarial tactics to maintain resilience.
[59]	Real Time Processing	• Achieving rapid detection and response to malware threats in real-time without compromising accuracy.	• Balancing speed and accuracy is challenging; complex models may introduce latency, while lightweight models may lack precision.	• Optimize model architectures to enhance speed without sacrificing accuracy, especially for edge and IoT environments.
[59]	Model Interpretability	• Improving transparency in AI model decisions to foster trust and facilitate collaboration between AI systems and human analysts	• Many AI models, especially deep learning ones, are "black boxes" with limited insight into decision-making processes.	• Employ explainable AI techniques that provide interpretable insights, enabling effective human AI collaboration.
[59]	Scalability and Adaptability	• Ensuring models can adapt to evolving malware patterns and process large-scale data inputs for continued effectiveness	• Static training datasets and traditional methods may not adequately capture the dynamic nature of malware.	• Research self-learning systems and adaptive models that update with new threats, along with real-time threat intelligence integration.

6. Proposed Future Research Directions

While numerous challenges persist, several promising proposed research directions shown in table 11, can guide the future of AI driven malware detection. Researchers can develop innovative solutions that address current limitations by exploring emerging trends and interdisciplinary collaborations.

Table 11: Proposed Future Directions in AI Driven Malware Detection

Proposed Future Direction	Description	Key Advantages	Research Focus
Federated Learning for Decentralized Malware Detection	Decentralized model training across devices without sharing raw data	Preserves data privacy, reduces overfitting	Develop optimized federated frameworks for malware detection

Data Privacy	Maintains control over sensitive information, a critical need in cybersecurity	Data security, compliance with regulations	Explore privacy preserving techniques in federated learning
Diverse Data Sources	Utilizes data from multiple sources, improving model robustness	Enhances model generalization, reduces overfitting	Incorporate data diversity in federated models for robustness
Real Time Updates	Continuous model updates to adapt to emerging threats	Ensures current threat detection	Focus on real-time federated updates with minimal latency
Blockchain for Secure Model Updates	Leverages Blockchain for secure, decentralized tracking of model updates	Ensures integrity, prevents model poisoning	Design Blockchain integrated frameworks for secure model updates
Traceability	Creates an auditable record of model updates for tracking changes	Improves transparency, vulnerability identification	Research auditable and transparent Blockchain based model updates
Consensus Mechanisms	Uses consensus protocols to validate updates and maintain system integrity	Verifies legitimacy of updates	Explore consensus algorithms that enhance model security in malware detection
Explainable AI for Transparency and Trust	Improves model interpretability, aiding trust and collaboration in cybersecurity	Enhances decision transparency, human AI collaboration	Develop practical XAI methods suitable for real-world deployment
Visualization Techniques	Visualization of model predictions and feature importance for understanding decisions	Clarifies AI reasoning for security analysts	Create interactive visualizations that enhance interpretability
Rule Based Explanations	Provides interpretable rule based insights alongside AI outputs	Facilitates human oversight, improves transparency	Integrate rule based systems for explainable decision-making
User Centric Design	Tailors explainability features to meet user needs and preferences	Increases trust, improves usability	Engage with users to design intuitive explainability features
Interdisciplinary Collaborations	Combines insights from cybersecurity, ethics, HCI, and legal studies	Enables comprehensive, well rounded solutions	Foster interdisciplinary approaches for multifaceted challenges
Cybersecurity and AI Ethics	Investigates ethical implications, including bias and accountability	Promotes responsible AI deployment	Address ethical concerns like accountability and bias in AI driven detection

Human AI Interaction	Studies user interactions with AI to improve usability and trust in AI driven security systems	Enhances system usability, trust	Develop user centric interfaces that facilitate effective security operations
Legal Frameworks	Examines the legal aspects of AI driven malware detection, focusing on privacy, compliance, and liability	Ensures compliance, clarifies accountability	Research legal guidelines that support responsible and compliant AI use in cybersecurity

Conclusion

The landscape of cyber security is evolving at an unprecedented pace, driven by the increasing sophistication of malware and the corresponding demand for effective detection solutions. This review has provided a comprehensive examination of the advancements and challenges associated with AI and deep learning techniques in malware detection. Through our exploration of various methodologies, we have identified key findings that underscore the transformative role of AI in enhancing the capabilities of malware detection systems. The significance of AI and deep learning in advancing malware detection cannot be overstated. These technologies provide the necessary tools to adapt to the rapidly changing threat landscape, enabling organizations to identify and respond to potential threats more effectively. By harnessing the power of machine learning algorithms, security practitioners can enhance their capabilities to detect novel malware strains and mitigate risks associated with cyber threats. As the volume and complexity of malware continue to rise, the role of AI in automating and optimizing detection processes will become increasingly vital. The integration of deep learning techniques will facilitate the development of more sophisticated detection systems capable of handling vast datasets, improving the accuracy and speed of threat identification. Future research must prioritize addressing the challenges associated with real time detection and adversarial resilience. Achieving a balance between detection speed and accuracy remains a critical concern, particularly in resource constrained environments such as edge computing and IoT devices. As organizations increasingly rely on these technologies, developing lightweight models that maintain robust performance will be essential for effective deployment. In parallel, research efforts should focus on enhancing the adversarial resilience of AI driven malware detection systems. Ongoing studies must explore dynamic defensive strategies that can adapt to evolving adversarial techniques, ensuring that detection systems remain effective in the face of increasingly sophisticated attacks. Developing interpretability methods will also be crucial, enabling security analysts to understand and trust the decisions made by AI models. As the field of malware detection continues to evolve, fostering interdisciplinary collaborations will be vital for advancing research. Integrating insights from cyber security, AI ethics, and human computer interaction will facilitate the development of more holistic solutions that address the complexities of AI driven malware detection.

References

[1] A. Chernikova *et al.*, “Cyber Network Resilience Against Self-Propagating Malware Attacks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13554 LNCS, no. 2019, pp. 531–550, 2022, doi: 10.1007/978-3-031-17140-6_26.

[2] A. A. Al-Hashmi *et al.*, “Deep-Ensemble and Multifaceted Behavioral Malware Variant Detection Model,” *IEEE Access*, vol. 10, pp. 42762–42777, 2022, doi: 10.1109/ACCESS.2022.3168794.

[3] S. Razaulla *et al.*, “The Age of Ransomware: A Survey on the Evolution, Taxonomy, and Research Directions,” *IEEE Access*, vol. 11, pp. 40698–40723, 2023, doi: 10.1109/ACCESS.2023.3268535.

[4] M. Yousefi-Azar, L. G. C. Hamey, V. Varadharajan, and S. Chen, “Malytics: A malware detection scheme,”

IEEE Access, vol. 6, pp. 49418–49431, 2018, doi: 10.1109/ACCESS.2018.2864871.

[5] 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, 2022, doi: 10.3390/s22186766.

[6] N. Marastoni, R. Giacobazzi, and M. Dalla Preda, "Data augmentation and transfer learning to classify malware images in a deep learning context," *J. Comput. Virol. Hacking Tech.*, vol. 17, no. 4, pp. 279–297, 2021, doi: 10.1007/s11416-021-00381-3.

[7] J. A. Mata-Torres, E. Tello-Leal, J. D. Hernandez-Resendiz, and U. M. Ramirez-Alcocer, "Evaluation of Machine Learning Techniques for Malware Detection," *Intell. Syst. Ref. Libr.*, vol. 226, pp. 121–140, 2023, doi: 10.1007/978-3-031-08246-7_6.

[8] E. Raff, W. Fleshman, R. Zak, H. S. Anderson, B. Filar, and M. McLean, "Classifying Sequences of Extreme Length with Constant Memory Applied to Malware Detection," *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 11A, pp. 9386–9394, 2021, doi: 10.1609/aaai.v35i11.17131.

[9] S. Mohammadi and M. Babagoli, "A novel hybrid hunger games algorithm for intrusion detection systems based on nonlinear regression modeling," *Int. J. Inf. Secur.*, vol. 22, no. 5, pp. 1177–1195, 2023, doi: 10.1007/s10207-023-00684-0.

[10] K. Lee, J. Lee, S. Y. Lee, and K. Yim, "Effective Ransomware Detection Using Entropy Estimation of Files for Cloud Services," *Sensors*, vol. 23, no. 6, 2023, doi: 10.3390/s23063023.

[11] F. Aldauiji, O. Batarfi, and M. Bayousef, "Utilizing Cyber Threat Hunting Techniques to Find Ransomware Attacks: A Survey of the State of the Art," *IEEE Access*, vol. 10, pp. 61695–61706, 2022, doi: 10.1109/ACCESS.2022.3181278.

[12] B. Jin, J. Choi, J. B. Hong, and H. Kim, "On the Effectiveness of Perturbations in Generating Evasive Malware Variants," *IEEE Access*, vol. 11, no. 1, pp. 31062–31074, 2023, doi: 10.1109/ACCESS.2023.3262265.

[13] A. Youssef, M. Abdelrazek, and C. Karmakar, "Use of Ensemble Learning to Detect Buffer Overflow Exploitation," *IEEE Access*, vol. 11, no. June, pp. 52009–52025, 2023, doi: 10.1109/ACCESS.2023.3279280.

[14] C. Tsfaty and M. Fire, "Malicious source code detection using a translation model," *Patterns*, vol. 4, no. 7, p. 100773, 2023, doi: 10.1016/j.patter.2023.100773.

[15] E. Nowroozi, Abhishek, M. Mohammadi, and M. Conti, "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 2, pp. 1332–1344, 2023, doi: 10.1109/TNSM.2022.3225217.

[16] E. Alshahrani, D. Alghazzawi, R. Alotaibi, and O. Rabie, "Adversarial attacks against supervised machine learning based network intrusion detection systems," *PLoS One*, vol. 17, no. 10 October, pp. 1–14, 2022, doi: 10.1371/journal.pone.0275971.

[17] C. Bolchini, L. Cassano, A. Miele, and A. Toschi, "Fast and Accurate Error Simulation for CNNs Against Soft Errors," *IEEE Trans. Comput.*, vol. 72, no. 4, pp. 984–997, 2023, doi: 10.1109/TC.2022.3184274.

[18] S. Khan, J. C. Chen, W. H. Liao, and C. S. Chen, "Towards Adversarial Robustness for Multi-Mode Data through Metric Learning," *Sensors*, vol. 23, no. 13, pp. 1–18, 2023, doi: 10.3390/s23136173.

[19] Y. X. Zhang, H. Meng, X. M. Cao, Z. Zhou, M. Yang, and A. R. Adhikary, "Interpreting vulnerabilities of multi-instance learning to adversarial perturbations," *Pattern Recognit.*, vol. 142, 2023, doi: 10.1016/j.patcog.2023.109725.

[20] L. Almuqren *et al.*, "Sine-Cosine-Adopted African Vultures Optimization with Ensemble Autoencoder-Based Intrusion Detection for Cybersecurity in CPS Environment," *Sensors*, vol. 23, no. 10, pp. 1–19, 2023,

doi: 10.3390/s23104804.

[21] A. Hernandez-Suarez *et al.*, "ReinforSec: An Automatic Generator of Synthetic Malware Samples and Denial-of-Service Attacks through Reinforcement Learning," *Sensors*, vol. 23, no. 3, 2023, doi: 10.3390/s23031231.

[22] F. Demirkıran, A. Çayır, U. Ünal, and H. Dağ, "An ensemble of pre-trained transformer models for imbalanced multiclass malware classification," *Comput. Secur.*, vol. 121, pp. 1–38, 2022, doi: 10.1016/j.cose.2022.102846.

[23] Y. Wang *et al.*, "A Geometrical Approach to Evaluate the Adversarial Robustness of Deep Neural Networks," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 19, no. 5 s, 2023, doi: 10.1145/3587936.

[24] D. Demirci, N. Sahin, M. Sirlancis, and C. Acarturk, "Static Malware Detection Using Stacked BiLSTM and GPT-2," *IEEE Access*, vol. 10, pp. 58488–58502, 2022, doi: 10.1109/ACCESS.2022.3179384.

[25] H. Najafzadeh, "Machine Learning Architectured Lifecycle.pdf." AWS Architecture Blog, 2021. [Online]. Available: https://aws.amazon.com/blogs/architecture/introducing-the-new-aws-well-architected-machine-learning-lens/

[26] F. Ullah, S. Ullah, M. R. Naeem, L. Mostarda, S. Rho, and X. Cheng, "Cyber-Threat Detection System Using a Hybrid Approach of Transfer Learning and Multi-Model Image Representation," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155883.

[27] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J. Big Data*, vol. 10, no. 1, 2023, doi: 10.1186/s40537-023-00694-8.

[28] T. Sonnekalb, T. S. Heinze, and P. Mäder, *Deep security analysis of program code: A systematic literature review*, vol. 27, no. 1. 2022. doi: 10.1007/s10664-021-10029-x.

[29] B. Schiele, "CNN Architecture 0522." Online Website, pp. 1–123, 2014. [Online]. Available: https://evbn.org/convolutional-neural-network-architecture-cnn-architecture-1678014845/

[30] D. Dera, S. Ahmed, N. C. Bouaynaya, and G. Rasool, "TRustworthy Uncertainty Propagation for Sequential Time-Series Analysis in RNNs," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 2, pp. 882–896, 2024, doi: 10.1109/TKDE.2023.3288628.

[31] G. Salmela, Lauri & Tsipinakis, Nikolaos & Foi, Alessandro & Billet, Cyril & Dudley, John & Genty, "Predicting ultrafast nonlinear dynamics in fibre optics with a recurrent neural network.," *Nat. Mach. Intell.*, no. 3, pp. 1–11, doi: 10.1038/s42256-021-00297-z.

[32] H. Li, H. Ge, H. Yang, J. Yan, and Y. Sang, "An Abnormal Traffic Detection Model Combined BiIndRNN With Global Attention," *IEEE Access*, vol. 10, pp. 30899–30912, 2022, doi: 10.1109/ACCESS.2022.3159550.

[33] I. Eddahmani, C. H. Pham, T. Napoléon, I. Badoc, J. R. Fouefack, and M. El-Bouz, "Unsupervised Learning of Disentangled Representation via Auto-Encoding: A Survey," *Sensors*, vol. 23, no. 4, pp. 1–19, 2023, doi: 10.3390/s23042362.

[34] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, 2018, doi: 10.1109/MSP.2017.2765202.

[35] F. Sarhangian, R. Kashef, and M. Jaseemuddin, "Efficient Traffic Classification Using Hybrid Deep Learning," *15th Annu. IEEE Int. Syst. Conf. SysCon 2021 - Proc.*, p. 6, 2021, doi: 10.1109/SysCon48628.2021.9447072.

[36] L. Dhanya, R. Chitra, and A. M. Anusha Bamini, "Performance evaluation of various ensemble classifiers for malware detection," *Mater. Today Proc.*, vol. 62, no. July, pp. 4973–4979, 2022, doi:

10.1016/j.matpr.2022.03.696.

[37] G. Ngo, R. Beard, and R. Chandra, "Evolutionary bagging for ensemble learning," 2022.

[38] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00349-y.

[39] S. Patil *et al.*, "Improving the robustness of ai-based malware detection using adversarial machine learning," *Algorithms*, vol. 14, no. 10, 2021, doi: 10.3390/a14100297.

[40] L. Chen, C. Yang, A. Paul, and R. Sahita, "Towards resilient machine learning for ransomware detection," no. Ml.

[41] M. Vohra, A. Tiwari, P. Sharma, and A. Jayaswal, "Malware Detection Using Deep Learning," *Lect. Notes Electr. Eng.*, vol. 1079 LNEE, no. 1, pp. 215–231, 2023, doi: 10.1007/978-981-99-5997-6_19.

[42] M. Williams *et al.*, "Detection Entropy-Based Network Traffic Analysis for Efficient Ransomware Detection," 2024.

[43] S. Wang *et al.*, "Towards Accountable and Resilient AI-Assisted Networks: Case Studies and Future Challenges," *2024 Jt. Eur. Conf. Networks Commun. 6G Summit, EuCNC/6G Summit 2024*, no. June, pp. 818–823, 2024, doi: 10.1109/EuCNC/6GSummit60053.2024.10597060.

[44] N. Anandharaj, "AI-Powered Cloud Security: A Study on the Integration of Artificial Intelligence and Machine Learning for Improved Threat Detection and Prevention," *J. Recent Trends Comput. Sci. Eng.*, vol. 12, no. 2, pp. 21–30, 2024.

[45] K. Sathupadi, "AI-Based Intrusion Detection and DDoS Mitigation in Fog Computing : Addressing Security Threats in Decentralized Systems," 2023.

[46] Z. Wang, C., Zhang, Y., Ding, H., "Applied Mathematics and Nonlinear Sciences," *Appl. Math. Nonlinear Sci.*, vol. 8, no. 2, pp. 3383–3392, 2023.

[47] M. Lavreniuk and O. Novikov, "Malicious and benign websites classification using machine learning methods," *Theor. Appl. Cybersecurity*, vol. 2, no. 1, pp. 29–31, 2020, doi: 10.20535/tacs.2664-29132020.1.209434.

[48] N. Hussen, S. M. Elghamrawy, M. Salem, and A. I. El-Desouky, "A Fully Streaming Big Data Framework for Cyber Security Based on Optimized Deep Learning Algorithm," *IEEE Access*, vol. 11, pp. 65675–65688, 2023, doi: 10.1109/ACCESS.2023.3281893.

[49] I. A. Kandhro *et al.*, "Detection of Real-Time Malicious Intrusions and Attacks in IoT Empowered Cybersecurity Infrastructures," *IEEE Access*, vol. 11, no. January, pp. 9136–9148, 2023, doi: 10.1109/ACCESS.2023.3238664.

[50] R. Le Guillou *et al.*, "A novel framework for quantifying accuracy and precision of event detection algorithms in fes-cycling," *Sensors*, vol. 21, no. 13, pp. 1–13, 2021, doi: 10.3390/s21134571.

[51] B. Bischl *et al.*, "OpenML Benchmarking Suites," no. NeurIPS, pp. 1–14, 2017, [Online]. Available: http://arxiv.org/abs/1708.03731

[52] X. W. JUNGEUM KIM, "Robust sensible adversarial learning of deep neural networks for image classification b," 2018.

[53] Z. Ying and B. Wu, "NBA: defensive distillation for backdoor removal via neural behavior alignment," *Cybersecurity*, vol. 6, no. 1, 2023, doi: 10.1186/s42400-023-00154-z.

[54] Z. Tian, M. Zhuo, L. Liu, J. Chen, and S. Zhou, "Anomaly detection using spatial and temporal information in multivariate time series," *Sci. Rep.*, vol. 13, no. 1, pp. 1–12, 2023, doi: 10.1038/s41598-023-31193-8.

[55] A. Mustafa, S. H. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao, "Deeply Supervised Discriminative Learning for Adversarial Defense," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3154–3166, 2021, doi: 10.1109/TPAMI.2020.2978474.

[56] K. A. Dhanya *et al.*, "Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2756–2766, 2023, doi: 10.1109/JSYST.2023.3238678.

[57] M. Golmaryami, R. Taheri, Z. Pooranian, M. Shojafar, and P. Xiao, "SETTI: A Self-supervised AdvErsarial Malware DeTection ArchiTecture in an IoT Environment," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 18, no. 2, 2022, doi: 10.1145/3536425.

[58] T. Anastasiou *et al.*, "Towards Robustifying Image Classifiers against the Perils of Adversarial Attacks on Artificial Intelligence Systems," *Sensors*, vol. 22, no. 18, pp. 1–22, 2022, doi: 10.3390/s22186905.

[59] H. J. Lee and Y. M. Ro, "Robust Proxy: Improving Adversarial Robustness by Robust Proxy Learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, no. 8, pp. 4021–4033, 2023, doi: 10.1109/TIFS.2023.3288672.