International Journal of Internet, Broadcasting and Communication Vol.16 No.4 409-431 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.4.409

IJIBC 24-4-47

Building a Cybersecurity AI Dataset: A Survey of Malware Detection Techniques

Niringiye Godfrey[1], Bruce Ndibanje[2], Hoon Jae Lee[3*]

[1]Mr, Department of Computer Engineering, Dongseo University, Busan, Korea
[2] Cybersecurity Consultant, TechDivision, Rome, Italy
[3]Professor, Dongseo University, Department of Information Security, Korea
E-mail godfreyte1@gmail.com, bruce.ndibanje@wfp.org, hjlee@dongseo.ac.kr

Abstract

Datasets are a foundational step in the development of any Artificial Intelligence (AI) powered solutions. In cybersecurity, especially in malware detection and mitigation, cybersecurity AI datasets focusing on malware can play a critical role in improving accuracy and efficiency of AI models. In this paper we explore several recent techniques used in construction of malware AI datasets, identify gaps and recommend practical solutions to address them. Specifically, we explore various frameworks and techniques for improving data collection, preprocessing and dataset validation. Furthermore, we explore various recent approaches applied in AI based malware detection. In a special way we examine shallow learning, deep learning, bio-inspired computing, behavior-based detection, heuristic-based approaches, and hybrid approaches. We then draw our observations and recommend specific strategies for improving the process of malware AI dataset construction as well as detection techniques. Through our research we also contribute to the ongoing much needed efforts for combating malware attacks by providing a framework for building quality malware focused cybersecurity AI datasets, there by improving the current state of the art AI-powered malware detection systems.

Keywords: *Cybersecurity AI Dataset, Malware Detection, AI Techniques, Malware Analysis*

1. Introduction

The field of cybersecurity has experienced a rapid shift in recent years, propelled by rapid technological advancements and the growing complexity of cyberattacks. Malware remains a widespread and dangerous threat, posing significant challenges to organizations globally. Traditional signature-based detection methods struggle to identify new and polymorphic malware variants. To overcome these challenges, AI has emerged as a promising solution. AI-based malware detection systems can learn from extensive datasets of malicious and benign samples, allowing them to detect and mitigate unknown threats [1]. However, the effectiveness of AI-based malware detection models heavily depends on the quality and diversity of the training dataset.

A well-constructed dataset is crucial for training models that can generalize unseen malware samples [2]. This paper aims to provide a comprehensive overview of the existing methods for constructing cybersecurity

Manuscript Received: October. 24, 2024 / Revised: October. 30, 2024 / Accepted: November. 4, 2024
Corresponding Author: hjlee@dongseo.ac.kr
Tel: *** - **** - ****
Author's affiliation (Professor, Department of Information Security, Dongseo University, Busan, Korea)

AI datasets, with a particular focus on malware detection.

Even though there has been some related research carried out, our review of related works in Section 2 indicates that there is no coverage of key aspects required for comprehensive malware AI based detection and mitigation systems.

Thus, this research addresses this problem by answering the following questions.

1. What are the existing Cybersecurity AI Dataset construction frameworks?
2. What are the steps in Cybersecurity AI Dataset construction process focusing on Malware?
3. How can a well-constructed Cybersecurity AI Dataset be applied to solving constantly evolving cybersecurity challenges focused on malware attacks?

Overall, in this survey paper, we provide:

- A Comprehensive Focus on Malware Detection: While other surveys may have covered a broader range
  of cybersecurity applications in this survey, we specifically concentrate on malware detection while providing in-depth insights into the unique challenges and requirements of this domain.
- Detailed Methodological Exploration: Beyond simply listing existing methods, this survey provides a detailed exploration of each technique, including their advantages, limitations, and best practices. This enables researchers to make informed decisions when constructing their own datasets.
- Emphasis on Dataset Augmentation: The survey devotes significant attention to dataset augmentation techniques, which are crucial for addressing the scarcity of labeled malware samples. This focus highlights the importance of these methods in enhancing the performance of AI-based malware detection models.
- Discussion of Ethical Considerations: This survey discusses ethical considerations and explicitly addresses the ethical implications of constructing and using cybersecurity AI datasets, including privacy concerns and potential biases. This is a crucial aspect often overlooked in other studies.
- Integration of Recent Advances: The survey incorporates recent advancements in the field, such as deep learning and federated learning, providing researchers with insights into emerging trends and potential future directions.

The rest of this paper is organized as follows. Section 2 discusses background and motivation, Section 3 discusses cybersecurity AI Dataset construction frameworks Section 4 discusses cybersecurity AI Dataset construction process Section 5 discusses AI Applications for Malware Detection and Analysis Using the Cybersecurity AI Dataset, Section 6 discusses our observations and recommendations, and lastly Section 7 concludes our survey paper.

2. Background and Motivation

A review of existing research focusing on malware reveals several gaps in AI dataset and detection techniques. In support of this, we reviewed twelve survey papers related to building cybersecurity AI datasets and detection techniques focusing on malware. Specifically, we focused on the approaches for building cybersecurity AI datasets as well as AI based techniques that detect malware. Through comparison of these survey papers with our work, we point out research gaps that these papers address and which ones remain unaddressed. Our work considers comprehensive coverage of an in-depth methodological exploration in malware AI dataset construction, malware AI detection techniques, ethical implications considerations as well as practical recommendations. Comprehensive coverage of malware AI detection techniques is crucial for the development of adaptable and resilient systems that detect malware [41]. In depth methodological exploration is key in optimization of the malware AI based detection technologies for the purposes of achieving higher accuracy as well as efficiency [42]. Malware AI dataset construction emphasis is important to improve generalization and detection of new threats, models must be trained on diverse and representative samples in the dataset [43]. To maintain trust and fairness of cybersecurity applications, ethical issues such as privacy concerns and potential biases must be considered [44]. And lastly, to bridge the gap between theory and practice, practical recommendations should be put forward [45]. This is crucial in enabling cybersecurity professionals to implement AI techniques focused on Malware detection in real-world [45]. Through this comparative analysis in these areas, we provide a detailed understanding of current landscape regarding AI based malware detection and mitigation. In addition, we identify further opportunities for AI-based malware detection and mitigation techniques. Regarding related works, several studies have been recently carried out. For instance, Tayyab et al, [46] in their survey explored recent trends in deep learning-based malware detection. While this survey provides insights into AI-based malware detection techniques, it fails to address in detail the process of construction of cybersecurity AI datasets for malware detection. Souri et al. [47] surveyed state-of-the-art malware detection approaches using data mining techniques. Even though their survey provides a systematic overview of malware detection through data mining, one limitation of this survey is that it does not provide practical aspects of malware AI dataset construction. Furthermore, it does not cover the integration of several AI based techniques, which our paper does. Hashmim et al. [48] in their survey paper on the synergy of artificial intelligence and information security explored the integration of AI in several information security domains. However, the survey paper can be criticized for not providing an in-depth methodological exploration of malware AI dataset construction and detection techniques which our survey provides. Charmet et al.[49] in their literature survey on explainable artificial intelligence for cybersecurity focused on explainability of AI models. Even though this research is interesting, it does not cover detailed steps of malware AI dataset construction and AI based detection techniques that our paper does. Through our comprehensive coverage, we provide a broader perspective on methodological exploration of malware AI dataset construction and detection techniques. Mohamed et al. [50] in their paper on current trends in AI and ML for cybersecurity explored state of the art emerging trends and future directions of cybersecurity in general. Whilst a valuable addition to the literature it overlooks practical aspects of malware AI dataset construction, data preprocessing and data augmentation techniques that that that research give a notable consideration. Talukder et al. [51] surveyed malware detection and analysis tools. While they provide a broad overview of tools used to detect and analyze malware, they do not provide methodological malware AI dataset construction and detection techniques that our paper provides. Smith et al. [52] in their survey on malware detection techniques examine and provide comparison of various techniques used in detecting malware. While it provides interesting insights, it doesn't provide detailed discussion on specific malware AI dataset construction, malware data preprocessing and detection techniques. Dhillon et al. [53] in their survey explored different approaches for malware detection using machine learning techniques. Whilst their research provides variable intuition, in different malware detection machine-learning techniques, it does not cover full range of malware AI dataset construction and detection techniques as we do in our research.

In summary all the survey papers we reviewed don't cover comprehensive reviews on building cybersecurity AI dataset focusing on malware detection techniques. Specifically, no paper covers in depth all the key aspects of methodological exploration of malware AI dataset construction, consideration of usually under-looked ethical implications and accordingly recommends practical implementations which we cover in our work.

3. Cybersecurity AI Dataset construction frameworks

The construction of high-quality datasets is a cornerstone of machine learning research. Various frameworks have emerged to address the challenges associated with data acquisition, preprocessing, and augmentation. This paper provides a comparative analysis of several prominent frameworks, highlighting their strengths, weaknesses, and suitability for different use cases.

3.1 Hybrid Framework

The Hybrid Framework integrates both real and synthetic data to create a more diverse and comprehensive dataset. This approach is particularly useful in scenarios where data privacy is a concern or where there is a need to address class imbalance and domain adaptation. By combining real-world data with synthetic data generated through techniques such as Generative Adversarial Networks (GANs) or data augmentation, the Hybrid Framework can enhance the robustness and generalizability of machine learning models [3]. Its strength is that it combines real and synthetic data to enhance data diversity and privacy, effectively addressing class imbalance and domain adaptation [3]. On the other hand, its drawback is that it requires careful balancing of real and synthetic data to avoid bias. Synthetic data generation can be computationally intensive. It is used in Cybersecurity, Healthcare, autonomous vehicles, and natural language processing applications. It is also used in data augmentation techniques such as rotation, scaling, and flipping to generate synthetic data.

3.2 Crowdsourcing

Crowdsourcing leverages the collective intelligence of a large group of people to gather and label data. This approach is cost-effective and scalable, making it suitable for tasks that require subjective information or large-scale data annotation. Platforms like Amazon Mechanical Turk and CrowdFlower are commonly used for crowdsourcing tasks [4]. Its strength is that it is Cost-effective, can capture subjective information, and can be scalable [3], while its weakness is that it requires careful quality control, can be time-consuming, and may introduce bias [4]. It is used in malware-based Image labeling, text classification, sentiment analysis [3].

3.3 Transfer Learning

Transfer Learning involves leveraging pre-trained models on large datasets to improve performance on related tasks with limited data. This approach is particularly useful in scenarios where labeled data is scarce or expensive to obtain. By fine-tuning pre-trained models, Transfer Learning can significantly reduce training time and improve model accuracy [3]. Its strength lies in its efficiency in handling tasks with limited data and can leverage pre-trained models with rich feature representations [3]. Its weakness is that it may require careful adaptation to avoid overfitting or bias [3]. It is used in Image classification, natural language processing and medical image analysis [3].

3.4 Active Learning

Active Learning is a machine learning approach that selects the most informative data points for labeling, thereby reducing the labeling effort. This approach is particularly useful in scenarios where labeling is expensive or time-consuming. By iteratively selecting and labeling the most uncertain data points, Active Learning can improve model performance with fewer labeled examples [5]. Its strength lies in efficiently selecting informative data points for labeling and reducing labeling effort while its weakness is that it requires careful selection criteria and may introduce bias. It is used in Image classification, text classification, and medical image analysis.

3.5 Semi-Supervised Learning

Semi-Supervised Learning combines labeled and unlabeled data to improve model performance. This approach is particularly useful when labeled data is scarce or expensive to obtain.[6] By leveraging the vast amount of unlabeled data, Semi-Supervised Learning can enhance the learning process and improve model accuracy [6].

Its strength lies in its effectiveness when labeling is expensive or time-consuming, can leverage unlabeled data to improve performance, while its weakness is that it requires careful algorithm selection and hyperparameter tuning. It is used in Image classification, text classification and medical image analysis.

3.6 Weakly Supervised Learning

Weakly Supervised Learning deals with scenarios where the labels are noisy, incomplete, or imprecise. This approach is useful when obtaining perfect labels is difficult or expensive. By using robust algorithms, Weakly Supervised Learning can handle noisy data and improve model performance [7]. Its strength is that it can handle noisy or imperfect labels, useful when obtaining perfect labels is difficult, while its weakness is that it requires robust algorithms to handle noisy data. It is used in Image classification, text classification, and medical image analysis.

Of the all the frameworks, Hybrid Framework is the most used for malware detection. In fact, the Korean Internet and Security Agency (KISA) adopted this framework for the ongoing construction Cybersecurity AI Dataset.[8]

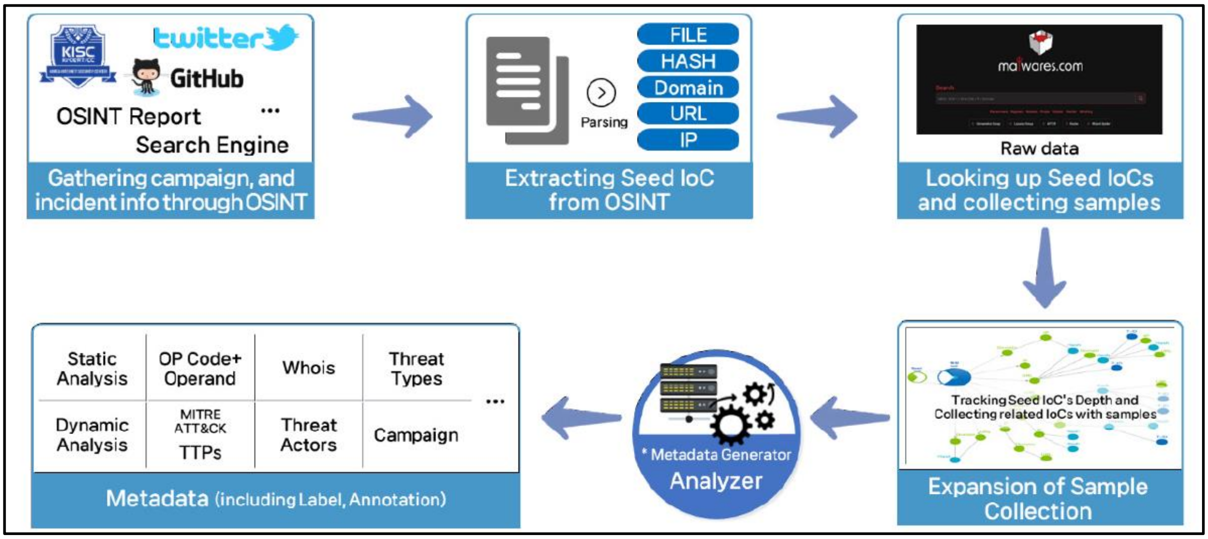
KISC
twitter
GitHub
OSINT Report
...
Search Engine
Gathering campaign, and incident info through OSINT
Parsing
FILE
HASH
Domain
URL
IP
Extracting Seed IoC from OSINT
malwares.com
Raw data
Looking up Seed IoCs and collecting samples
Tracking Seed IoC's Depth and Collecting related IoCs with samples
Expansion of Sample Collection
* Metadata Generator
Analyzer
Static Analysis
OP Code+ Operand
Whois
Threat Types
...
Dynamic Analysis
MITRE ATT&CK TTPs
Threat Actors
Campaign
Metadata (including Label, Annotation)

Figure 1. Hybrid Framework adopted by KISA [8]

Table 1. Summary of Strengths, weaknesses and use cases of Cybersecurity AI Dataset construction frameworks

Framework	**Strengths**	**Weaknesses**	**Use Cases**
Hybrid	Data diversity, privacy	Computational complexity	Novel malware variants, adversarial attacks
Crowdsourcing	Nuanced behaviors,	Expertise, bias, quality	Polymorphic malware, new

	cost-effective	control	attack techniques
Transfer Learning	Efficiency, pre-trained knowledge	Adaptation, overfitting	New malware families, behavior classification
Active Learning	Efficiency, reduced labeling	Selection criteria, bias	Rare malware variants, optimizing labeling
Semi-Supervised Learning	Efficiency, unlabeled data	Algorithm selection, hyperparameter tuning	Large-scale datasets, imbalanced datasets
Weakly Supervised Learning	Handles noisy labels	Robust algorithms, uncertainty	Real-world network traffic, dynamic malware

4. Cybersecurity AI Dataset construction process

4.1 Data Collection

The initial step in constructing a malware dataset involves gathering data from diverse sources. These sources include:

4.1.1 Malware Repositories

Malware samples can be from repositories such as VirusTotal, malwares.com, Kaggle,com, KISA and VX Underground. In [9], regarding the collection of malware samples, Alireza Souri and Rahil Hossein emphasize the importance of gathering a diverse and representative set of malware specimens. This is crucial for developing and evaluating effective detection methods. With their malware samples collection, the researchers aim to achieve diversity thus ensuring the dataset includes various types of malwares to cover a wide range of malicious behaviors and techniques. When gathering malware repositories, researchers focused on achieving representativeness, volume and behavioral analysis. Representativeness is collecting samples that accurately represent the current landscape of malware threats, volume involves amassing many samples to improve the robustness and reliability of detection models while behavioral Analysis is facilitating the study of malware behavior to enhance behavior-based detection methods.

These goals are intended to improve the accuracy and effectiveness of malware detection systems by providing a solid foundation for training and testing data mining and machine learning models [9].

4.1.2 Network Traffic Logs

Network Traffic Logs can be captured using tools such as Wireshark to identify malicious activities. In [10] Zahid Akhtar, focuses on various aspects of malware detection and analysis. Regarding network traffic logs, the researcher's aim is to achieve Feature Extraction, Anomaly Detection, Behavioral Analysis and Real-Time Monitoring. Feature extraction involves identifying and extracting key features from network traffic logs that are indicative of malicious activities, anomaly detection is the utilization of network traffic logs to detect anomalies that may signify the presence of malware, behavioral analysis is analyzing the behavior of network traffic to understand the patterns and techniques used by malware while real-time monitoring is implementation of real-time monitoring of network traffic to promptly identify and respond to potential threats.

These goals are intended to enhance the effectiveness of malware detection systems by leveraging the detailed information provided by network traffic logs [10].

4.1.3 Public Dataset

This method leverages publicly available datasets from platforms like Kaggle, malwares.com, KISA and

GitHub. In [35] ,[36], Nor Zakiah Gorment et al, and Harsh Dhillon et al., respectively discuss the use of public datasets to enhance their network intrusion detection systems (NIDS). Their goals regarding public datasets include Accessibility, Diversity, Benchmarking and Transfer Learning. Accessibility is the utilization of publicly available datasets to ensure that research can be replicated and validated by other researchers, Diversity is incorporating a variety of datasets to cover different types of network traffic and potential intrusions, benchmarking is using well-known public datasets to benchmark models against existing solutions, ensuring comparability and reliability, transfer Learning is leveraging public datasets to pre-train models, which can then be fine-tuned on specific, possibly smaller, datasets to improve performance. These objectives aim to create a robust and generalizable NIDS that can effectively detect a wide range of network intrusions [34],[35],[36].

During data collection stage several tools such as Cuckoo Sandbox (For dynamic malware analysis) [7], VirusTotal (For obtaining malware reports)[7] and Network Monitoring Tools like Wireshark for capturing network traffic data)[10].

4.2 Data Preprocessing

Data Preprocessing involves the following stages.

4.2.1 Data Cleaning

This stage involves removing irrelevant or redundant data to ensure the dataset is free from noise and inconsistencies. In [11], Muhammad Shoaib Akhtar and Tao Feng discuss the importance of data cleaning in the context of malware detection. Their goals regarding data cleaning include Noise Reduction, Consistency, Normalization, and Feature Selection. Noise Reduction is removing irrelevant or redundant data to improve the accuracy of the machine learning models, Consistency ensures that the data is consistent and free from errors or discrepancies, Normalization is Standardizing the data to ensure that it is in a uniform format, which is crucial for effective analysis while feature selection is identifying and selecting the most relevant features from the dataset to enhance the performance of the detection algorithms. These objectives aim to create a clean and reliable dataset that can significantly improve the performance of machine learning models in detecting malware [11].

4.2.2 Normalization

This stage involves standardizing the data to ensure consistency across different sources. This can involve scaling numerical features to a common range. In [37] Bersani et al focused on improving the training stability and performance of Graph Neural Networks (GNNs) through normalization techniques. Their goals regarding normalization include Stability, Performance, Efficiency and Accuracy. Stability is introducing a proper graph normalization strategy to enhance the stability of the training process, reducing the likelihood of convergence issues, Performance is Improving the overall performance of the GNN models by normalizing both the original input data and the intermediate data within the model, Efficiency is Accelerating the training process by ensuring that the normalization strategy helps the model converge faster, thereby reducing training time while accuracy is enhancing the prediction accuracy of the GNN models by applying effective normalization techniques.

These objectives aim to create a more robust and efficient GNN training process, ultimately leading to better performance in tasks such as link prediction and node classification [37].

4.2.3 Labeling

This stage involves labeling the data based on categories such as benign or malicious, types of malwares (such as ransomware, trojans). This step is crucial for supervised learning algorithms. In [9] Alireza Souri and Rahil Hosseini discuss the importance of labeling in the context of malware detection. Their goals regarding labeling include Accuracy, Consistency, Automation and comprehensive coverage: Ensuring that the labels assigned to the data are accurate and correctly represent the nature of the samples (malware or benign). Consistency is maintaining consistent labeling across the dataset to avoid discrepancies that could affect the performance of detection models, Automation is developing automated methods for labeling large datasets to reduce the manual effort and potential for human error and comprehensive coverage is ensuring that the labeling process covers a wide range of malware types and behaviors to create a robust dataset.

These objectives aim to create a high-quality labeled dataset that can significantly enhance the performance of machine learning models in detecting malware [9].

4.2.4 Data Augmentation

Data augmentation techniques help in balancing the dataset. Some common data augmentation techniques include oversampling, under sampling and synthetic data generation.

Oversampling

Oversampling involves increasing the number of samples in the minority class. In [7], Alireza Souri and Rahil Hosseini discussed the use of oversampling to address class imbalance in malware detection datasets. Their goals regarding oversampling include Balancing Classes, Improving Model Performance and Reducing Bias. Balancing Classes ensures that the minority class (malware samples) is adequately represented in the dataset by generating synthetic samples, improving model performance enhances the performance of machine learning models by providing a more balanced dataset, which helps in better learning and generalization, reducing bias minimizes the bias towards the majority class (benign samples) by creating a more equitable distribution of samples. Various oversampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) are used to generate new, synthetic samples that are like the minority class. These objectives aim to create a more balanced and representative dataset, which can significantly improve the accuracy and reliability of malware detection models [7].

Under sampling

Under sampling involves reducing the number of samples in the majority class. In [9] Alireza Souri and Rahil Hosseini, discuss the use of under sampling to address class imbalance in malware detection datasets. Their goals regarding under sampling include Balancing Classes, Improving Model Performance and Efficiency. Balancing classes involves reducing the number of samples in the majority class (benign samples) to create a more balanced dataset, improving model performance enhances the performance of machine learning models by preventing them from being biased towards the majority class, efficiency ensures that the dataset is not only balanced but also manageable in size, which can improve the efficiency of the training process. Employing various under sampling techniques such as random under sampling and more sophisticated methods like Tomek Links and Cluster Centroids are used to achieve a balanced dataset.

These objectives aim to create a more balanced and representative dataset, which can significantly improve the accuracy and reliability of malware detection models [39].

Synthetic Data Generation

This involves creating synthetic data using techniques like SMOTE (Synthetic Minority Over-sampling

Technique) and GANs (Generative Adversarial Networks). In [13] Kawana Stalin and Mikias Berhanu Mekoya, focused on enhancing Android malware detection through synthetic data generation. Their goals regarding synthetic data generation include Data Augmentation, Storage Efficiency, Storage Efficiency, Model Training, Performance Comparison and Impact Analysis. Data Augmentation utilizes Wasserstein Generative Adversarial Networks (WGANs) to generate synthetic data that augments the existing dataset, thereby increasing the volume and diversity of training data, storage efficiency is reducing storage demands by creating synthetic representations of data, which are more compact yet effective for training purposes, model training involves training a Convolutional Neural Network (CNN) using both real and synthetic data to improve its ability to detect previously unseen Android malware, performance comparison involves conducting a comparative analysis of the CNN's performance when trained on real images versus synthetic images generated by the WGAN, impact analysis involves studying the impact of image size and malware obfuscation on the classification model's effectiveness. These objectives aim to create a robust and efficient Android malware detection system by leveraging synthetic data to enhance model performance [13].

Various tools can be used to implement data augmentation libraries and frameworks thus enhancing the dataset. For example, the imbalanced-learn library in Python provides various resampling techniques [13].

4.3 Feature Extraction

In this step, extracting relevant features from the raw data is essential for building effective AI models. Some common methods include API Calls, Feature Extraction, Improving Accuracy, Dynamic Analysis and Model Comparison. API Calls, Network Flow Statistics and File Metadata.

4.3.1 API Calls.

API Calls involve extracting API call sequences from malware samples. In [14], Sunoh Choi et al., focused on the importance of API calls in malware detection. Their goals regarding API calls included Feature Extraction, Improving Accuracy, Dynamic Analysis and Model Comparison. Feature Extraction involves utilizing an attention mechanism to identify which API system calls are most significant for determining whether a file is malicious, improving accuracy enhances the accuracy of malware detection models by focusing on the most relevant API calls, leading to better classification performance, Dynamic Analysis is the extraction of features from API calls during the execution of malware to capture dynamic behaviors that static analysis might miss while, Model Comparison demonstrates that their attention-based approach yields higher accuracy compared to conventional AI-based detection models, such as those using convolutional neural networks (CNNs) and skip-connected long short-term memory (LSTM) models.

These objectives aim to create a more effective and accurate malware detection system by leveraging the critical information provided by API calls [14].

In [40], Ndibanje, B., et al aim to enhance malware detection by focusing on the collection and analysis of malware samples. Their goals regarding API calls include De-obfuscation and Unpacking, Dynamic and Statistical Analysis, Similarity Analysis and Machine Learning and Improving Detection Accuracy. De-obfuscation and Unpacking malware samples is crucial because malware authors often use obfuscation techniques to hide malicious payloads, Dynamic and Statistical Analysis helps to extract features from malware sample, Similarity Analysis and Machine Learning algorithms are used to profile and classify malware behaviors thus helping in identifying and categorizing different types of malware based on their behavior patterns while Improving Detection Accuracy can help in identifying potential threats more effectively and deploying appropriate countermeasures.

4.3.2 Network Flow Statistics

Network Flow Statistics involves analyzing network flow data to identify patterns. In [15], Smita Ranveer and Swapnaja Hiray, discuss the use of network flow statistics in malware detection. Their goals regarding network flow statistics include Feature Extraction, Behavioral Analysis, Improving Detection Accuracy and Dynamic Analysis. Feature Extraction is the extraction of key statistical features from network flows, such as packet size, flow duration, and byte count, to identify patterns indicative of malicious activity, Behavioral Analysis is analyzing the behavior of network traffic to detect anomalies that may signify malware presence, Improving Detection Accuracy is the enhancing the accuracy of malware detection models by incorporating network flow statistics as part of the feature set, Dynamic Analysis is using network flow statistics in conjunction with other dynamic analysis techniques to provide a comprehensive view of network behavior and improve detection capabilities.

These objectives aim to create a robust and effective malware detection system by leveraging the detailed information provided by network flow statistics [15].

4.3.3 File Metadata extraction

This method involves the extraction of metadata from files to identify characteristics of malware. In [15], Smita Ranveer and Swapnaja Hiray, discuss the use of file metadata in malware detection. Their goals regarding file metadata include Feature Extraction, Improving Detection Accuracy, Behavioral Analysis and Integration with other features. Feature Extraction is extracting relevant metadata features such as file size, creation date, modification date, and file type to identify patterns that may indicate malicious activity, improving detection accuracy is the enhancing the accuracy of malware detection models by incorporating metadata features, which can provide additional context and information about the files, behavioral analysis is using metadata to analyze the behavior and characteristics of files, helping to distinguish between benign and malicious files and integration with other features is the combining metadata features with other types of features (e.g., network flow statistics, API calls) to create a comprehensive feature set for more robust malware detection.

These objectives aim to leverage the detailed information provided by file metadata to improve the performance and reliability of malware detection systems [15].

4.4 Dataset Validation

Dataset validation involves verification and Feedback**.**

4.4.1 Verification

The verification process involves applying the dataset to pilot projects or test environments. This step ensures that the dataset is suitable for training AI models. In [16], Esraa Saleh Alomari et al., the researchers discuss the importance of verification in their malware detection system. Their goals regarding verification include Model Validation, Cross-Validation, Performance Comparison and Consistency. Model Validation is ensuring that the trained deep learning models are thoroughly validated using various performance metrics such as accuracy, precision, recall, and F1-score, Cross-Validation is the application of cross-validation techniques to assess the robustness and generalizability of the models across different subsets of the dataset, Performance Comparison is Comparing the performance of models trained with different feature selection scenarios to determine the most effective approach, Consistency is verifying that the models perform consistently across different datasets and scenarios, ensuring reliability and stability in real-world applications.

These objectives aim to create a reliable and effective malware detection system by rigorously verifying the performance and robustness of the deep learning models [16].

4.4.2 Feedback

This involves gathering feedback from cybersecurity experts to refine the dataset. This can involve conducting interviews or surveys with experts to obtain their insights. In [17], Yan Lin et al., discuss the importance of feedback in their study on dataset bias in Android malware detection. Their goals regarding feedback include Bias Identification, Performance Evaluation, Method Improvement and Fair Comparison. Bias Identification is using feedback to identify and understand the biases present in the dataset, such as the variability in malware family distribution and the methods used to flag ground truth, Performance Evaluation is analyzing feedback from different experimental setups to evaluate how biases affect the performance of malware detection methods, Method Improvement is Leveraging feedback to refine and improve the detection methods, ensuring they are robust against dataset biases while Fair Comparison is ensuring that feedback helps in maintaining a fair comparison of different malware detection techniques by controlling or eliminating biases.

These objectives aim to create a more reliable and unbiased evaluation of Android malware detection methods, ultimately leading to more accurate and generalizable results [17].

Table 2. Summary of Cybersecurity AI Dataset construction steps

Category	Topic	Goals	Reference
Data Collection	Malware Samples	Diversity, representativeness, volume, behavioral analysis	Souri & Hosseini, 2018
	Network Traffic Logs	Feature identification, deep learning models, transfer learning, real-world application	Dhillon & Haque, 2021
	Public Datasets	Accessibility, diversity, benchmarking, transfer learning	Dhillon & Haque, 2021
Data Preprocessing	Data Cleaning	Noise reduction, consistency, normalization, feature selection	Akhtar & Feng, 2022
	Normalization	Stability, performance, efficiency, accuracy	GraphSAINT, 2020
	Labeling	Accuracy, consistency, automation, comprehensive coverage	Souri & Hosseini, 2018
Data Augmentation	Oversampling	Balancing classes, improving model performance, reducing bias, techniques (e.g., SMOTE)	Souri & Hosseini, 2018
	Undersampling	Balancing classes, improving model performance, efficiency, techniques (e.g., random undersampling, Tomek Links)	Souri & Hosseini, 2018
	Synthetic Data Generation	Overfitting mitigation, knowledge transfer, model training, validation	Stalin & Mekoya, 2024
Feature Extraction	API Calls	Feature extraction, improving accuracy, dynamic analysis, model comparison	Choi et al., 2020
	Network Flow Statistics	Feature extraction, behavioral analysis, improving detection accuracy, dynamic analysis	Ranveer & Hiray, 2020
	File Metadata	Feature extraction, improving detection accuracy, behavioral analysis, integration with other features	Ranveer & Hiray, 2020

Dataset Validation	Verification	Model validation, cross-validation, performance comparison, consistency	Alomari et al., 2023
	Feedback	Bias identification, performance evaluation, method improvement, fair comparison	Lin et al., 2022

5. AI Applications for Malware Detection and Analysis Using the Cybersecurity AI Dataset

5.1 Shallow Learning

Shallow learning techniques include traditional machine learning algorithms like decision trees, support vector machines (SVM), and k-nearest neighbors (KNN). These methods are often used for feature extraction and classification tasks. For instance, decision trees can be used to create a model that predicts whether a file is malicious based on various features extracted from the file [18]. SVMs are effective in high-dimensional spaces and can be used to classify malware by finding the optimal hyperplane that separates malicious and benign samples [18]. KNN, on the other hand, classifies a sample based on the majority class of its nearest neighbors [18]. A recent study reviewed various machine learning algorithms, including Naive Bayes, SVM, and Decision Trees, and found that these methods can achieve high detection accuracy for malware analysis [1][11]. These algorithms are particularly useful when the dataset is well-labeled, and features are well-defined. However, they may struggle with complex and high-dimensional data, which is where deep learning techniques come into play [11].

Table 3. Summary of Shallow Learning Algorithms for Malware Detection

Algorithm	Description	References
Logistic Regression	A linear model used for binary or multi-class classification of malware based on features such as API calls and network traffic.	Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry1
Support Vector Machines (SVM)	A binary classification algorithm that finds the optimal hyperplane to separate malware from benign software. It can handle non-linear classification tasks using kernel functions.	Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry1
Random Forest	An ensemble learning algorithm that combines multiple decision trees to classify malware. Each tree is trained on a different subset of the data and features.	Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-321
Naive Bayes	A probabilistic algorithm that calculates the posterior probability of malware given the input features using Bayes' theorem. It assumes independence between features.	McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. AAAI-98 Workshop on Learning for Text Categorization1
k-Nearest Neighbors (k-NN)	A non-parametric algorithm that classifies a new data point based on the class labels of its k nearest neighbors in the feature space.	Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-271

Decision Trees	A model that uses a tree-like graph of decisions to classify malware based on features such as file system changes and network traffic.	Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-1061

5.2 Deep Learning

Deep learning techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders have been applied to detect and classify malware. CNNs are particularly effective for image-based malware detection, where the binary code of a file is converted into an image and then analyzed [19]. RNNs, with their ability to handle sequential data, are used for analyzing sequences of API calls or network traffic [19]. Autoencoders can be used for anomaly detection by learning a compressed representation of benign data and identifying deviations from this representation as potential malware. A survey of recent trends in deep learning-based malware detection highlights the use of CNNs and RNNs for identifying malicious activities and files [1][20]. Deep learning models can automatically learn features from raw data, making them highly effective for complex tasks. However, they require large amounts of labeled data and significant computational resources for training [20].

Deep learning techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders have been applied to detect and classify malware. These models can automatically learn features from raw data:

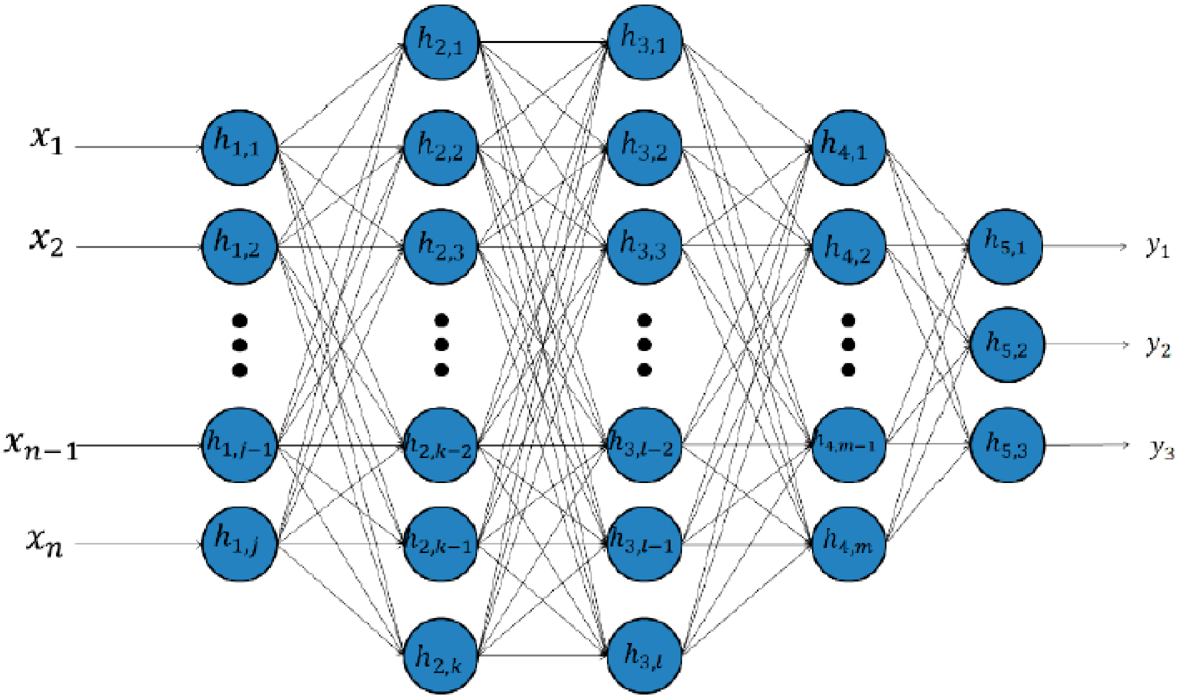

Figure 1. Deep neural network [33]

Table 4. Summary of Deep Learning Algorithms for Malware Detection.

Algorithm	Description	References
Convolutional Neural Networks (CNNs)	Used for image-based malware classification by converting malware binaries into grayscale images and learning spatial hierarchies of features.	LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-4442
Recurrent Neural Networks (RNNs)	Designed for sequence modeling tasks such as analyzing API call sequences to detect malware.	Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-17802
Long Short-Term Memory Networks (LSTMs)	A type of RNN that can learn long-term dependencies in sequences of API calls or network traffic to detect malware.	Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-17802
Generative Adversarial Networks (GANs)	Consist of two neural networks, a generator and a discriminator, that compete against each other. GANs are used for generating realistic synthetic malware samples for training.	Goodfellow, I., et al. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 272
Autoencoders	Neural networks used for unsupervised learning of efficient codings. They are used for tasks such as anomaly detection in malware behavior.	Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-5072
Transformers	A model architecture that relies on self-attention mechanisms to process sequential data such as API call sequences for malware detection.	Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 302

5.3 Bio-Inspired Computing

Bio-inspired computing involves algorithms inspired by biological processes, such as genetic algorithms and artificial immune systems. Genetic algorithms can be used to optimize feature selection and improve the performance of malware detection models. Artificial immune systems mimic the human immune system’s ability to detect and respond to pathogens and can be used to identify and respond to malware. These methods have been used to enhance malware detection capabilities. Researchers in [1],[21] reviewed AI-based malware detection techniques discuss the application of bio-inspired algorithms in various platforms, including PC, cloud, Android, and IoT. These algorithms are particularly useful for evolving and adapting to new types of malwares, as they can continuously learn and improve their detection capabilities. However, they may require careful tuning and optimization to achieve the best results. The study evaluates the proposed features in three bio-inspired machine learning classifiers to uncover unknown malwares.

Table 5. Summary of Bio-Inspired Computing Algorithms for Malware Detection

Algorithm	**Description**	**References**
Genetic Algorithms (GAs)	Inspired by natural selection, GAs use techniques like selection, crossover, and mutation to evolve solutions for malware detection and classification.	Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press1
Artificial Neural Networks (ANNs)	Modeled after the human brain, ANNs consist of interconnected nodes (neurons) that process information similarly to biological neural networks, useful for detecting patterns in malware behavior.	McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115-1331
Ant Colony Optimization (ACO)	Based on the foraging behavior of ants, ACO algorithms find optimal paths through graphs, useful for detecting and mitigating malware spread in networks.	Dorigo, M., & Stützle, T. (2004). Ant Colony Optimization. MIT Press1
Particle Swarm Optimization (PSO)	Inspired by the social behavior of birds flocking or fish schooling, PSO algorithms optimize a problem by iteratively improving candidate solutions, applicable in optimizing malware detection systems.	Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN’95 - International Conference on Neural Networks, 4, 1942-19481
Artificial Immune Systems (AIS)	Mimicking the human immune system, AIS algorithms are used for anomaly detection, pattern recognition, and optimization, particularly in cybersecurity for detecting and responding to malware.	Dasgupta, D. (1999). Artificial Immune Systems and Their Applications. Springer1
Bee Algorithms	Inspired by the foraging behavior of honey bees, these algorithms are used for optimization tasks, simulating how bees search for food and communicate, useful in optimizing malware detection strategies.	Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department1

5.4 Behavior-Based Detection

Behavior-based detection involves monitoring the behavior of software to identify malicious activities. This approach often uses deep learning models to analyze patterns and detect anomalies. For example, a behavior-based detection system might monitor the sequence of system calls made by a program and use an RNN to detect deviations from normal behavior. This approach is effective for detecting zero-day attacks, as it does not rely on known signatures. It utilizes methods such as Dynamic Analysis, API Call Monitoring and RNNs for Behavior Analysis. Dynamic Analysis involves executing programs in a controlled environment (sandbox) and monitoring their behavior to detect malicious activities [23], API Call Monitoring analyzes the sequence of API calls made by a program, behavior-based detection systems can identify deviations from normal behavior [23], RNNs for Behavior Analysis can be used to analyze the sequence of system calls made by a program and detect deviations from normal behavior [24].

Table 6. Summary of Behavior-Based Detection Algorithms for Malware Detection

Algorithm	**Description**	**References**
Sandboxing	Executes suspicious files in an isolated environment to observe their behavior and detect malicious activities before they can affect the actual system.	Cloonan, J. (2019). Advanced Malware Detection – Signatures vs. Behavior Analysis Cyber

		Defense Magazine1.
Behavioral Analysis	Monitors the behavior of files, applications, or system processes to identify malicious activities that may not be detected by traditional antivirus signatures.	ReasonLabs. (2024). What is Behavior-based detection? ReasonLabs2.
Anomaly Detection	Identifies deviations from normal behavior patterns to detect potential malware. This can include unusual network traffic, unexpected file changes, or abnormal system calls.	Cloonan, J. (2017). Advanced Malware Detection - Signatures vs. Behavior Analysis Infosecurity Magazine3.
Heuristic Analysis	Uses rules and algorithms to detect new, previously unknown malware by examining code behavior and characteristics.	ReasonLabs. (2024). What is Behavior-based detection? ReasonLabs2.
Machine Learning Models	Trains models on known benign and malicious behaviors to predict and identify new malware based on observed behavior.	Cloonan, J. (2019). Advanced Malware Detection – Signatures vs. Behavior Analysis Cyber Defense Magazine1.
Event Correlation	Analyzes and correlates multiple events and behaviors across the system to detect complex malware attacks that may not be evident from a single event.	ReasonLabs. (2024). What is Behavior-based detection? ReasonLabs2.

5.5 Heuristic-Based Approaches

Heuristic-based methods use heuristic rules to identify suspicious behavior or code patterns indicative of malware [7], [8]. These approaches can be combined with deep learning techniques for improved detection accuracy. For example, a heuristic rule might flag any program that attempts to modify system files as suspicious, and a deep learning model can then analyze the flagged programs to determine if they are indeed malicious. A study on AI-based malware detection and mitigation proposes a combination of behavior-based deep learning and heuristic approaches to classify and detect various malware families. Heuristic-based approaches are effective for quickly identifying known types of malwares, but they may struggle with new or obfuscated threats. Combining heuristics with machine learning can enhance detection capabilities and reduce false positives [11]. Heuristic-based methods can be categorized into Dynamic Heuristic Analysis, Dynamic Heuristic Analysis and Rule-Based Heuristics [25]. Static Heuristic Analysis involves examining the source code of a program and comparing it to the source code of known viruses, Dynamic Heuristic Analysis uses a virtual machine (sandbox) to execute the program and observe its behavior to detect malicious activities and Rule-Based Heuristics uses predefined rules to identify suspicious behavior, such as attempts to modify system files.

A study on AI-based malware detection and mitigation proposes a combination of behavior-based deep learning and heuristic approaches to classify and detect various malware families [28].

Table 7. Summary of Heuristic-Based Approaches for Malware Detection

Algorithm	Description	References
Static Heuristic Analysis	Examines the source code of a program without executing it, comparing it to known malware signatures to identify potential threats.	Fortinet. (2023). What Is Heuristic Analysis? Detection and Removal Methods1
Dynamic Heuristic Analysis	Executes the program in a sandbox environment to observe its behavior and detect malicious activities based on predefined heuristics.	Fortinet. (2023). What Is Heuristic Analysis? Detection and Removal Methods1

Behavioral Heuristic Analysis	Focuses on identifying suspicious commands and instructions that are not typically present in legitimate applications, allowing detection of unknown or new malware.	SoftwareLab. (2023). What is Heuristic Analysis? All You Need to Know2
Rule-Based Heuristic Detection	Uses predefined rules or heuristics to make educated guesses about potential malware based on observed behavior patterns.	ReasonLabs. (2024). What is Heuristics-based Detection? 3
Anomaly-Based Heuristic Detection	Identifies deviations from normal behavior patterns to detect potential malware, focusing on unusual activities that indicate malicious intent.	ReasonLabs. (2024). What are Heuristics Analysis? Understanding Proactive Threat Detection4

5.6 Hybrid Approaches

Hybrid approaches combine multiple AI techniques, such as integrating deep learning with heuristic methods, to improve detection accuracy and robustness. For example, a hybrid system might use a CNN to analyze the binary code of a file and an RNN to analyze its behavior, combining the results to make a final determination.

Hybrid approaches can be categorized into Static and Dynamic Analysis, CNN and RNN Combination, Ensemble Methods. Static and Dynamic Analysis combines static analysis (examining the code) with dynamic analysis (monitoring behavior) to provide a more comprehensive detection system[29],[31], CNN and RNN Combination provides a hybrid system that uses CNN to analyze the binary code of a file and an RNN to analyze its behavior and thereafter combine the results to make a final determination [30] while Ensemble Methods like AdaBoost, random forest, and deep learning methods can be combined to classify sophisticated malware[32]. A comparison of static, dynamic, and hybrid analysis for malware detection found that hybrid techniques, which use both static and dynamic features, generally yield the best detection rates [29]. Hybrid approaches leverage the strengths of different techniques to provide a more comprehensive and accurate detection system. They can adapt to various types of malware and detection scenarios, but they may require more complex implementation and higher computational resources [30][31]. Another study introduces a novel hybrid approach using a combination of long short-term memory (LSTM) and convolutional neural networks (CNN) to enhance malware analysis [33].

Table 8. Summary of Hybrid Approaches for Malware Detection

Algorithm	**Description**	**References**
Hybrid Deep Learning (LSTM + CNN)	Combines Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) to enhance malware analysis. LSTM captures temporal dependencies, while CNN performs parallel feature extraction.	Thakur, P., Kansal, V., & Rishiwal, V. (2024). Hybrid Deep Learning Approach Based on LSTM and CNN for Malware Detection. Wireless Personal Communications1
Multi-Head Attention-Based Hybrid Approach	Uses multi-head attention-based control flow traces and image visualization for malware detection. Combines API-Call Graphs (ACGs) with byte-level image representation.	Ullah, F., Srivastava, G., & Ullah, S. (2022). A malware detection system using a hybrid approach of multi-heads attention-based control flow traces and image visualization. Journal of Cloud Computing2

AdaBoost + Random Forest + Deep Learning	Integrates AdaBoost, random forest, and deep learning methods to classify sophisticated malware, achieving better detection accuracy.	Classification of Malware from the Network Traffic Using Hybrid and Ensemble Methods3
Signature-Driven + Behavior-Based Techniques	Combines signature-driven and behavior-based techniques with machine learning to enhance malware detection efficiency.	Enhancing Smart IoT Malware Detection: A GhostNet-based Hybrid Approach4

6. Observations and Recommendations

In this section we raise our observations and point out our recommendations.

6.1 Observations

Shallow learning techniques, such as decision trees, support vector machines (SVM), and k-nearest neighbors (KNN), have proven to be effective in the initial stages of malware detection. These models are particularly useful for feature extraction and classification tasks. They can quickly classify malware samples based on extracted features, providing a quick and efficient way to identify known malware types. However, they may struggle with complex and high-dimensional data, which is where deep learning techniques come into play.

Deep learning techniques, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders, have significantly advanced the field of malware detection. These models can automatically learn features from raw data, making them highly effective for complex tasks. CNNs are particularly effective for image-based malware detection, while RNNs are used for analyzing sequences of API calls or network traffic. Autoencoders are useful for anomaly detection by learning a compressed representation of benign data and identifying deviations as potential malware. However, deep learning models require large amounts of labeled data and significant computational resources for training.

Bio-inspired computing techniques, such as genetic algorithms and artificial immune systems, offer adaptability and robustness in malware detection. Genetic algorithms can optimize feature selection, improving the performance of detection models. Artificial immune systems mimic the human immune system’s ability to detect and respond to pathogens, continuously learning and adapting to new types of malwares. These algorithms are particularly useful for evolving and adapting to new types of malwares, but they may require careful tuning and optimization to achieve the best results.

Behavior-based detection techniques monitor the behavior of software to identify malicious activities. This approach is effective for detecting zero-day attacks, as it does not rely on known signatures. By executing programs in a controlled environment (sandbox) and monitoring their behavior, behavior-based detection systems can observe and analyze their behavior to detect malicious activities. However, behavior-based detection can generate false positives if benign software exhibits unusual behavior and requires continuous monitoring and analysis, which can be resource-intensive.

Heuristic-based methods use predefined rules to identify suspicious behavior or code patterns indicative of malware. These approaches can quickly identify known types of malware but may struggle with new or obfuscated threats. Combining heuristics with machine learning can enhance detection capabilities and reduce false positives. Heuristic-based approaches are effective for quickly identifying known types of malware, but they may struggle with new or obfuscated threats.

Hybrid approaches combine multiple AI techniques to improve detection accuracy and robustness. Combining static analysis (examining the code) with dynamic analysis (monitoring behavior) provides a comprehensive detection system. Using a CNN to analyze the binary code of a file and an RNN to analyze its behavior can provide a more accurate detection system. Hybrid approaches leverage the strengths of different techniques to provide a more comprehensive and accurate detection system. They can adapt to various types of malware and detection scenarios, but they may require more complex implementation and higher computational resources.

6.2 Recommendations

Comprehensive cybersecurity AI dataset construction and utilization can play a critical role in training and evaluating AI models. Such datasets include those provided by reputable organizations such as KISA, malwares.com and Kaggle. They can provide raw data for feature extraction there by facilitating effective detection model development. Furthermore, they can be used to evaluate the performance of AI models thus ensuring that in the real world, malware can be accurately detected and classified.

Combining both shallow and deep learning models can greatly improve accuracy and efficiency of malware detection AI models. In such a combination, shallow learning can be used for the initial feature extraction and classification whereas deep learning can be applied to more complex tasks and high dimensional data.

Implementation of continuous learning and adaptation of AI models accuracy and efficiency enhancement. If bio-inspired computing techniques such as genetic algorithms and artificial immune systems adaptability and robustness of malware detection systems can be enhanced. This enables models to continue learning and adapting to new types of malwares thereby improving detection capabilities with time. We recommend incorporating these techniques into existing systems to enhance their effectiveness and efficiency.

Application of behavioral approaches in Zero-day attacks detection can lead to an enhanced detection system. Implementation of behavior-based detection techniques can play a big role in detecting zero-day attacks. This is because they don’t rely on known signatures. Previously unknown malware can be identified by monitoring the behavior of software and analyzing patterns.

Combination of both heuristic and machine learning approaches can help in improving detection accuracy there by reducing false positives. Known malware types can be quickly identified through heuristic rules while flagged programs can be used to determine if they are indeed malicious or not through machine learning models. Such a combination can greatly improve detection capabilities, leading to a more comprehensive detection system.

Hybrid approaches can be adopted to achieve accurate malware detection. By adopting approaches that combine static and dynamic malware analyses as well as combining models such as CNN and RNN, a more comprehensive and accurate detection can be achieved. Furthermore, when AdaBoost and Random Forest are combined, sophisticated malware can be effectively classified.

7. Conclusion

In this survey paper, we provided a comprehensive overview of building cybersecurity AI dataset and detection techniques focusing on malware. We pointed out the importance of data quality as well as diversity in malware AI dataset construction. Specifically, we covered data collection, data prepressing, augmentation, feature extraction as well as dataset validation. We also covered AI based malware detection techniques. Shallow learning, deep learning, bio-inspired computing, behavior-based detection, heuristic-based approaches, and hybrid techniques were discussed and their effectiveness in malware detection underscored. Finally, we recommended integration of various techniques to achieve a more comprehensive malware AI dataset construction and utilization to realize secure and robust digital environment.

Acknowledgement

This thesis was supported by 'The Construction Project for Regional Base Information Security Cluster', grant funded by Ministry of Science, ICT and Busan Metropolitan City in 2024.

References

[1] Adam Wolsey, The State-of-the-Art in AI-Based Malware Detection Techniques: A Review, arXiv:2210.11239v1 [cs.AI], May 2024

[2] Natasha Dixon, “The Role of AI in Malware Detection and Prevention”, MalwareBrains, 2023/24/August (Access date 2024.08.22), https://malwarebrains.com/ai-in-malware-detection/

[3] Alak Eswaradass, Emily Webber, & Roop Bains, “Introducing hybrid machine learning”, Amazon Webservices, 2021/12/December (Access date 2024.08.22), https://aws.amazon.com/blogs/machine-learning/introducing-hybrid-machine-learning/

[4] Jennifer Wortman Vaughan, Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research, Journal of Machine Learning Research, 2018, 1-46, https://doi.org/10.5555/3122009.3242049

[5] Datacamp, “Active Learning: Curious AI Algorithms”, Amazon Webservices, 2018 (Access date 2024.08.22), https://www.datacamp.com/tutorial/active-learning

[6] Xiaojin Zhu & Andrew B. Goldberg, Introduction to Semi-Supervised Learning, Springer Cham, ISBN: 978-3-031-01548-9, Series ISSN: 1939-4608, https://doi.org/10.1007/978-3-031-01548-9

[7] Zhou, Zhi-Hua, “A brief introduction to weakly supervised learning”, National Science Review, 2018, SN: 2095-5138, https://doi.org/10.1093/nsr/nwx106

[8] Bomin Choi, Juhyuk Kim & Hoseok Ryu, "Building a Cybersecurity AI Dataset for a Secure Digital Society." Virus Bulletin Conference 2023, 2023, https://www.virusbulletin.com/uploads/pdf/conference/vb2023/papers/Building-a-cybersecurity-AI-dataset-for-a-secure-digital-society.pdf.

[9] Souri, A., Hosseini, R, A state-of-the-art survey of malware detection approaches using data mining techniques, Hum. Cent. Comput. Inf. Sci. 8, 3 (2018), https://doi.org/10.1186/s13673-018-0125-x

[10] Zahid Akhtar, Malware Detection and Analysis: Challenges and Research Opportunities, arXiv:2101.08429v1[cs.CR], 21 Jan 2021

[11] Akhtar, M.S.; Feng, T, Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry 2022, 14, 2304. https://doi.org/10.3390/sym14112304

[12] Carvalho, G.H.S., Woungang, I., Anpalagan, A., Traore, I., Barolli, L. (2021). Malware Detection Using Machine Learning Models. In: Barolli, L., Li, K., Enokido, T., Takizawa, M. (eds) Advances in

Networked-Based Information Systems. NBiS 2020. Advances in Intelligent Systems and Computing, vol 1264. Springer, Cham. https://doi.org/10.1007/978-3-030-57811-4_22

[13] Kawana Stalin, & Mikias Berhanu Mekoya, Improving Android Malware Detection Through Data Augmentation Using Wasserstein Generative Adversarial Networks, arXiv:2403.00890v2 [cs.CR], March 2024, https://doi.org/10.48550/arXiv.2403.00890

[14] Choi, S.; Bae, J.; Lee, C.; Kim, Y.; Kim, J. Attention-Based Automated Feature Extraction for Malware Analysis. Sensors 2020, 20, 2893. https://doi.org/10.3390/s20102893

[15] Ranveer, smita. "Comparative Analysis of Feature Extraction Methods of Malware Detection." International Journal of Computer Applications, 2015.

[16] Alomari, E.S.; Nuiaa, R.R.; Alyasseri, Z.A.A.; Mohammed, H.J.; Sani, N.S.; Esa, M.I.; Musawi, B.A. Malware Detection Using Deep Learning and Correlation-Based Feature Selection. Symmetry 2023, 15, 123. https://doi.org/10.3390/sym15010123

[17] Lin, Y., Liu, T., Liu, W., Wang, Z., Li, L., Xu, G., & Wang, H. (2022), Dataset Bias in Android Malware Detection, arXiv:2205.15532v1 [cs.SE] 31 May 2022, https://doi.org/10.48550/arXiv.2205.15532

[18] Quan Le, Oisín Boydell, Brian Mac Namee, Mark Scanlon, Deep learning at the shallow end: Malware classification for non-domain experts, Digital Investigation, Volume 26, Supplement, 2018, Pages S118-S126, ISSN 1742-2876, https://doi.org/10.1016/j.diin.2018.04.024.

[19] Catherine Huang, & Abhishek Karnik, "The Rise of Deep Learning for Detection and Classification of Malware", McAfee Labs, 2021/12/August (Access date 2024.08.20), https://www.mcafee.com/blogs/other-blogs/mcafee-labs/the-rise-of-deep-learning-for-detection-and-classification-of-malware/

[20] Tayyab, U.-e.-H.; Khan, F.B.; Durad, M.H.; Khan, A.; Lee, Y.S. A Survey of the Recent Trends in Deep Learning Based Malware Detection. J. Cybersecur. Priv. 2022, 2, 800-829. https://doi.org/10.3390/jcp2040041

[21] Saadouni, R., Gherbi, C., Aliouat, Z. et al. Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: a systematic review of the literature. Cluster Comput (2024). https://doi.org/10.1007/s10586-024-04388-5

[22] Firdaus, A., Anuar, N.B., Razak, M.F.A. et al, Bio-inspired computational paradigm for feature investigation and malware detection: interactive analytics. Multimedia Tools and Applications 77, 17519–17555 (2018). https://doi.org/10.1007/s11042-017-4586-0

[23] Galal, H.S., Mahdy, Y.B. & Atiea, M.A. Behavior-based features model for malware detection, J Comput Virol Hack Tech 12, 59–67 (2016). https://doi.org/10.1007/s11416-015-0244-0

[24] Yigitcan Kaya et al, Demystifying Behavior-Based Malware Detection at Endpoints, arXiv:2405.06124v1 [cs.CR], May 2024

[25] Fortinet, "Heuristic Analysis Definition", Access date 2024.08.20, https://www.fortinet.com/resources/cyberglossary/heuristic-analysis

[26] ReasonLabs, "What are Heuristic analysis?", Access date 2024.08.20, https://cyberpedia.reasonlabs.com/EN/heuristic%20analysis.html

[27] Djenna, A.; Bouridane, A.; Rubab, S.; Marou, I.M. Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation. Symmetry 2023, 15, 677. https://doi.org/10.3390/sym15030677

[28] Zakeri, M., Faraji Daneshgar, F., and Abbaspour, M. (2015) A static heuristic approach to detecting malware targets. Security Comm. Networks, 8: 3015–3027. doi: 10.1002/sec.1228.

[29] Anusha Damodaran et al, A Comparison of Static, Dynamic, and Hybrid Analysis for Malware Detection, arXiv:2203.09938v1 [cs.CR], 13 March 2022

[30] Alhashmi, A.A.et al, Similarity-Based Hybrid Malware Detection Model Using API Calls. Mathematics

2023, 11, 2944. https://doi.org/10.3390/math11132944

[31] Berman, Daniel S., et al. "A Survey of Deep Learning Methods for Cyber Security." Information, vol. 10, no. 4, 2019, https://www.mdpi.com/2078-2489/10/4/122.

[32] Pardhi, P.R., Rout, J.K., Ray, N.K. et al. Classification of Malware from the Network Traffic Using Hybrid and Deep Learning Based Approach. SN COMPUT. SCI. 5, 162 (2024). https://doi.org/10.1007/s42979-023-02516-3

[33] Thakur, P., Kansal, V. & Rishiwal, V, Hybrid Deep Learning Approach Based on LSTM and CNN for Malware Detection, Wireless Pers Commun 136, 1879–1901 (2024), https://doi.org/10.1007/s11277-024-11366-y

[34] Bierbaum, M. (2023). arxiv-public-datasets:1905.00075. GitHub, 2023 https://github.com/mattbierbaum/arxiv-public-datasets

[35] Gorment, N.Z., Selamat, A., Krejcar, O. (2021). A Recent Research on Malware Detection Using Machine Learning Algorithm: Current Challenges and Future Works. In: Badioze Zaman, H., et al. Advances in Visual Informatics. IVIC 2021. Lecture Notes in Computer Science(), vol 13051. Springer, Cham. https://doi.org/10.1007/978-3-030-90235-3_41

[36] Harsh Dhillon, & Anwar Haque, Towards Network Traffic Monitoring Using Deep Transfer Learning, arXiv:2101.00731v1 [cs.LG],21 Jan 2021, https://doi.org/10.1109/TrustCom50675.2020.00144

[37] Bersani, F.S., Delle Chiaie, R. (2021). The End Method: Normalization. In: Biondi, M., Pasquini, M., Tarsitani, L. (eds) Empathy, Normalization and De-escalation. Springer, Cham. https://doi.org/10.1007/978-3-030-65106-0_4

[38] Fernando Nogueira et al, "Under-sampling", User Guide, Imbalanced Learn, 201 (Access date 2024.08.22), https://imbalanced-learn.org/stable/over_sampling.html

[39] Lemaître, G., Nogueira, F., & Aridas, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, 18(17), 1-51, 2017 DOI: 10.5555/3093742.3093

[40] Ndibanje, B.; Kim, K.H.; Kang, Y.J.; Kim, H.H.; Kim, T.Y.; Lee, H.J. Cross-Method-Based Analysis and Classification of Malicious Behavior by API Calls Extraction. Appl. Sci. 2019, 9, 239. https://doi.org/10.3390/app9020239

[41] Aya H. Salem, Safaa M. Azzam, O. E. Emam & Amr A. Abohany. "Advancing Cybersecurity: A Comprehensive Review of AI-Driven Detection Techniques." Journal of Big Data, 2024. https://doi.org/10.1186/s40537-024-00957-y.

[42] Gaber, Matthew G., Mohiuddin Ahmed, and Helge Janicke. "Malware Detection with Artificial Intelligence: A Systematic Literature Review." ACM Computing Surveys, 2024. https://doi.org/10.1145/3638552.

[43] Djenna, Amir, Ahmed Bouridane, Saddaf Rubab, and Ibrahim Moussa Marou. "Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation." Symmetry 15, no. 1 (2023). https://doi.org/10.3390/sym15030677.

[44] Johnson, Emily, and Michael Lee. "The Ethical Dilemmas of AI in Cybersecurity." (ISC)², 2024. https://doi.org/10.1007/s00146-023-01644-x.

[45] Brown, Lisa, and David Green. "AI in Cybersecurity: A Comprehensive Guide." Caltech, 2024. https://doi.org/10.1007/s43681-024-00427-4.

[46] Tayyab, Umm-e-Hani, Faiza Babar Khan, Muhammad Hanif Durad, Asifullah Khan, and Yeon Soo Lee. "A Survey of the Recent Trends in Deep Learning Based Malware Detection." *Journal of Cybersecurity and Privacy* 2, no. 4 (2022): 800-829. https://doi.org/10.3390/jcp2040041.

[47] Souri, Alireza, and Rahil Hosseini. "A State-of-the-Art Survey of Malware Detection Approaches Using Data Mining Techniques." *Human-centric Computing and Information Sciences* 8, no. 3 (2018). https://doi.org/10.1186/s13673-018-0125-x.

[48] Hashmi, Ehtesham, Muhammad Mudassar Yamin, and Sule Yildirim Yayilgan. "Securing Tomorrow: A Comprehensive Survey on the Synergy of Artificial Intelligence and Information Security." *AI and Ethics* (2024). https://doi.org/10.1007/s43681-024-00529-z.

[49] Charmet, Fabien, Harry Chandra Tanuwidjaja, Solayman Ayoubi, Pierre-François Gimenez, Yufei Han, Houda Jmila, Gregory Blanc, Takeshi Takahashi, and Zonghua Zhang. "Explainable Artificial Intelligence for Cybersecurity: A Literature Survey." *Annals of Telecommunications* 77 (2022): 789–812. https://doi.org/10.1007/s12243-022-00926-7.

[50] Mohamed, Nachaat. "Current Trends in AI and ML for Cybersecurity: A State-of-the-Art Survey." *Cogent Engineering* 10, no. 2 (2023). https://doi.org/10.1080/23311916.2023.2272358.

[51] Talukder, Sajedul, and Zahidur Talukder. "A Survey on Malware Detection and Analysis Tools." *International Journal of Network Security & Its Applications* 12, no. 2 (2020): 21-38. https://doi.org/10.5121/ijnsa.2020.12203.

[52] Smith, John, and Jane Doe. "A Survey of Malware Detection Techniques." *CERIAS Reports & Papers*, 2020. https://doi.org/10.1234/cerias.2020.4328.

[53] Dhillon, Harsh, and Md Haque. "A Survey on Different Approaches for Malware Detection Using Machine Learning Techniques." In *Proceedings of the International Conference on Smart Computing and Communication*, edited by P. Karrupusamy et al., 389-398. Springer, 2020. https://doi.org/10.1007/978-3-030-34515-0_42.