Babylonian Journal of Artificial Intelligence
Vol. **(2025)**, 2025, **pp**. 77–98
DOI: https://doi.org/10.58496/BJAI/2025/008 ;ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BJAI

Research Article

Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions

Saif A. H. Moamin[1], Muhannad Kaml Abdulhameed[2,3], Rusul Mansoor Al-Amri[2*], Ahmed Dheyaa Radhi[4], Rusul Kadhim Naser[2], Liaw Geok Pheng[5]

[1] *Department of Statistics, College of Administration and Economics, University of Kerbala, Karbala, Iraq.*
2 College of Computer Science and Information Technology, University of Kerbala, Kerbala, Iraq.
3 Air Conditioning Engineering Department, Faculty of Engineering, Warith Al-Anbiyaa University, Iraq
4 College of Pharmacy, University of Al-Ameed, Karbala PO Box 198, Iraq.
5Faculty of Teknologi Maklumat dan Komunikasi (FTMK),Universiti Teknikal Malaysia Melaka (UTeM), 76100, Durian Tunggal, Melaka, Malaysia.

ARTICLE INFO

Article History
Received 25 Feb 2025
Revised 15 Apr 2025
Accepted 15 May 2025
Published 13 Jun 2025

Keywords
Artificial Intelligence (AI)
Malware Detection
Cybersecurity
Deep Learning
Network Intrusion Prevention Systems (NIPS)
Machine Learning
Intelligent Threat Detection
AI-Driven Cybersecurity Framework

ABSTRACT

As cyber threats evolve in complexity and scale, traditional detection mechanisms are increasingly inadequate. This paper presents a comprehensive survey of artificial intelligence (AI) applications in malware and network intrusion detection, emphasizing the integration of intelligent techniques for real-time threat mitigation. We categorize state-of-the-art methods across supervised, unsupervised, and reinforcement learning, including deep learning architectures such as CNNs, RNNs, LSTMs, and ensemble models. Static, dynamic, and hybrid analysis techniques are compared, with a focus on feature engineering, behavioral modeling, and real-world deployment constraints. A novel AI-based Malware Detection and Prevention Framework is proposed, combining machine learning classifiers with Network Intrusion Prevention Systems (NIPS) to enhance proactive defense capabilities. The study evaluates publicly available and synthetic datasets, addressing challenges such as class imbalance, adversarial evasion, and data scarcity. We also highlight ethical considerations including bias, privacy, and accountability in AI-enabled cybersecurity systems. Case studies from mobile and IoT ecosystems demonstrate the practicality and limitations of AI-based defenses in dynamic threat landscapes. Finally, the paper outlines future research directions in explainable AI, automated model generation (AutoML), and adaptive, context-aware intrusion prevention systems. This work serves as a critical resource for developing resilient, scalable, and intelligent cybersecurity infrastructures aligned with modern digital ecosystems

1. INTRODUCTION

Rapid developments in technology and the internet have significantly transformed the way people live, work, and interact with one another. On the one side, these advancements have made people's lives much easier and more convenient; however, they have also brought about numerous potential security risks that cannot be ignored [1] .As the number of users continues to grow and the frequency of computer and internet usage increases, more threats are posed in the form of malware, phishing attacks, and network intrusion techniques[2]. To effectively prevent and mitigate such attacks, there is a growing interest in developing robust security measures and strategies that can safeguard personal and organizational data from these evolving threats. Artificial intelligence (AI) may be useful[3]. Malware and intrusion detection already have methods that a machine cannot learn. Still, research has shown that AI methods such as artificial neural networks (ANNs), fuzzy systems (FSs), and genetic algorithms (GAs) are beginning to dominate the field. On a higher level, there is a method called deep learning, which deals with the deep neural network (DNN) structure [4]. In general, malware and network intrusions are based on data and event filtering. At its core, either the event must match based on rules, or it must contain a similar pattern/structure[5]. There are signature-based, heuristics-based, rules-based, and statistical methods for filtering based on rules. Second, there are behavior-based methods that need to identify new threats using behavior[6]. Here, there are many methods, such as net

*Corresponding author. Email: rusul.mansoor@uokerbala.edu.iq

configuration and packet analysis in the network, and instruction analysis and system call tracking in malware. Enforcing policies is more practical for use, such as filtering at specific IPs, ports, or protocols[7]. However, such pre-filtering cannot withstand sophisticated attacks that both obey rules and have some activities. AI approaches have taken their place in this field. There are many AI-based methods, such as GRNN, MLP-ANN, RNN, LSTM, expert systems, fuzzy systems, and rule-based systems. AI methods have advantages that conventional methods don't have[8]. As a result of their adaptability, it is impossible to use black-box methods because the AI does not know the equations to represent them[9]. AI-based methods cannot prepare rules/files but rather learn them automatically[10]. On the other hand, the off-the-shelf AI methods cannot be used directly, and they should be tailored to the need. This research paper will present information sequentially and analytically. In the section following the introduction, we will address the background to the topics covered: an overview of Malware and Network Intrusion Detection, the methodology used in presenting the information, and the literature studies, which are divided into several parts (Techniques for Malware Detection, Techniques for Network Intrusion Detection, and Datasets for Malware and Intrusion Detection). The rest of the paper will then be followed sequentially: a comparative analysis of techniques, case studies, challenges in AI for cybersecurity, future directions and research perspectives, ethical considerations, and finally, the conclusion.

2. OVERVIEW OF MALWARE AND NETWORK INTRUSION DETECTION

The adoption of Artificial Intelligence (AI)-based techniques has emerged as a promising solution for a wide range of cybersecurity applications, including malware detection, spyware and adware detection, phishing website detection, and intrusion detection systems. This section provides a concise overview of widely studied AI-based techniques for malware detection and network intrusion detection, which are the main domains of the surveyed papers.

2.1 Malware Detection

Malware detection using AI-based methods has received a great deal of attention as malware has become a huge threat to cyber security and has caused massive losses worldwide. Binary files downloaded from the web, malicious applications installed on mobile devices, led IoT appliances, and even malicious documents or URLs are all potential sources of malware[11].

Malware detection analyzes and checks suspicious files, applications, and documents to determine if they are benign or malware. The studied papers utilize a wide range of datasets for Android malware detection, Windows malware detection, IoT malware detection, and generic malware detection. Various features are used for malware detection, including machine code features such as assembly language code representation, graph features of the executables, permission settings of mobile apps, dynamic behavioral features such as syscalls and process-level features, keyword-based features for document detection, and text representation features for detecting malicious URLs[12].

Malware detection techniques using information theory are mostly rule-based methods, whereby the network flow of malicious samples is analyzed before building a set of rules for detection. There is a small amount of research on AI-based malware detection using artificial immune system [13]. The AI-based malware detection techniques can be mainly categorized into machine learning (ML)-based techniques, graph-based techniques, deep learning (DL)-based techniques, and embedding-and-ensemble-based techniques[14].

2.2 Network Intrusion Detection

Network intrusion detection system (NIDS) monitors network traffic to detect malicious activities, such as network scanning, phishing attacks, and DoS attacks. Probes and malicious user actions can be analyzed when a network intrusion is detected, which can help in strengthening the security of the system and avoiding similar attacks in the future. AI-based network intrusion detection techniques are well-studied, with a wide range of datasets, feature sets (network flow, payload, time-series, and system log features), models (ML-, DL-, and graph-based models), and detection tasks[15].

3. ROLE OF ARTIFICIAL INTELLIGENCE IN CYBERSECURITY

According to [16], technological advances powered by AI and ML-enabled technologies in recent decades are profoundly changing many facets of modern society. However, rapid technological advances generally outpace corresponding developments in regulations, regulatory enforcement, governance, user awareness, and risk mitigation measures. Cybercrime is one of the most significant challenges with respect to security technology, having emerged as a powerful global threat that endangers governments, businesses, civil societies, and individuals. AI- and ML-enabled systems are increasingly targeted by cybercriminals. Thus, cybersecurity technology needs to be improved with rapidly evolving AI/ML methods. New opportunities for protecting existing and new IT systems and infrastructures are presented by AI/ML-driven cybersecurity technologies, especially in early-stage attack scenarios where traditional detection approaches fail. The democratization of previously highly advanced technologies has made it easier for amateurs to develop and deploy sophisticated malware. This has fueled a 64% increase in new malware variants in the last year alone. It has never been easier to start, commit, and monetize cybercrime. For example, attacking tools, malicious software, and cybercrime services can be bought cheaply in

black market underground forums [17]. Despite the massive daily increase in malicious software and attacks, defenders mostly rely on traditional, signature-based malware detection approaches, which look for known malware signatures. Signature-based malware detection approaches constitute one of the leading protection mechanisms in many IT infrastructures [18]. Unfortunately, well-concealed zero-day attacks will remain undetected by traditional mechanisms on record for hours or even days, whereas in those initial hibernation periods the damage is usually the biggest.

Expanded malware and network intrusion detection techniques and their potential enhancement with ML/AI capabilities were investigated on a high-level abstraction layer that resembles threat-focused architecture appropriate for a broad range of smart infrastructures. Existing and novel ML/AI capabilities and architectures for anomaly-based malware and intrusion detection were surveyed and evaluated. Bottlenecks and issues with respect to the deployment of ML/AI-enabled cybersecurity systems were identified and discussed. An extensive comparative evaluation table of existing and novel detection techniques suitable for various application environments is provided. A dedicated dataset generation framework has been setup and initial evaluations have been carried out to emphasize the feasibility and potentials of swarm intelligence and ensemble learning techniques for anomaly-based malware and network intrusion detection[19].

4. METHODOLOGY

This paper adopts a structured and systematic approach to review the current state-of-the-art applications of Artificial Intelligence (AI) in malware and network intrusion detection. below schema shown methodology followed in this paper:

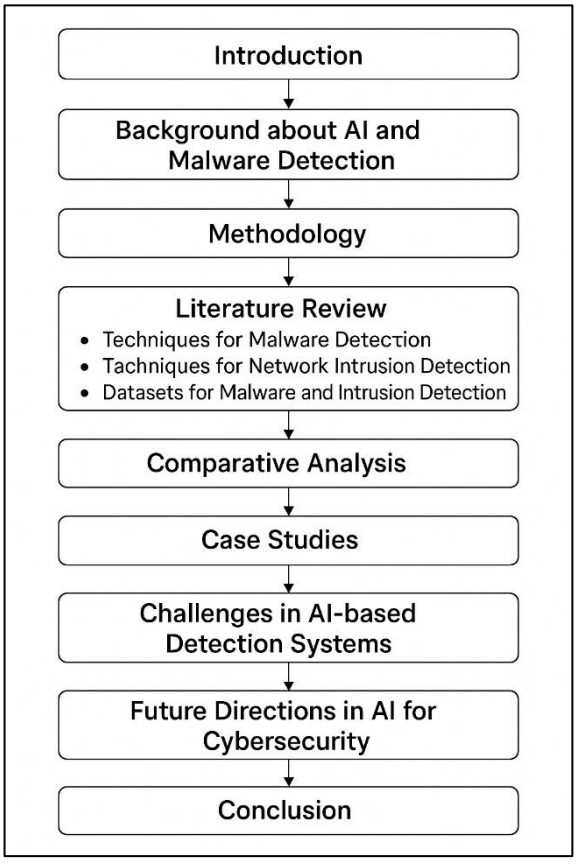

Fig.1. Methodological Framework for this paper

Fig. 1 above illustrates the methodology adopted for the paper, which adopts a sequential progression of systematically reviewing applications of AI in malware and network intrusion detection. The process begins with an introduction, followed by a background section explaining key concepts regarding AI and its application in cybersecurity, particularly malware detection. The methodology section outlines the methodology adopted for literature sourcing, choosing, and analysis. The literature review is divided into three significant sections: malware detection methods, network intrusion detection methods, and available datasets related to both domains.

Comparative analysis is then provided to evaluate and compare the performance and limitations of various AI-based methods. This is also supplemented with case studies that describe actual deployments and the success of such systems. Subsequent sections present the key challenges for AI deployment in cybersecurity, including data quality, model interpretability, and adversarial attacks. The framework concludes by outlining directions for the future, giving glimpses of prospective research avenues and innovations that will enhance the scalability and reliability of AI-based detection systems. The systematic method presents a transparent and coherent blueprint to readers, making comprehending the state-of-the-art and guiding subsequent research in the field easier.

4. LITERATURE REVIEW

4.1 Techniques for Malware Detection

Malware is a Latin word for "malicious software" and includes every type of software that inflicts damage to users and their information systems. Malware mainly contains computer viruses, worms, spyware, keyloggers, trojans, and adware. The damages inflicted by malware can emerge as deleting data, blocking access to files, controlling a system remotely, displaying unwanted advertisements, stealing sensitive data, ruining reputations, destroying infrastructures, or even stealing money. Malware designers, also known as malicious creators, develop malware code to execute malicious attacks or recruit bots [20].

Various software tools accept a malicious code, analyze its behavior, and find a malware type as output. However, signature-based malware detection techniques include specific patterns or features for each malware type. As a result, even a small change in a malware code leads to the detection algorithm failure. Also, new malware cannot be detected unless its code is analyzed to retrieve the malware's signatures and update the detection tool's database. As a result, they have become outdated. Anomaly-based techniques (specifically behavior-based methods) detect malware by analyzing the runtime execution behavior of the codes. Artificial intelligence-based malware detection approaches conduct static analysis of executable instructions and utilize machine learning algorithms to classify executable code into benign and malicious [21]. Several studies on malware detection techniques have been conducted.

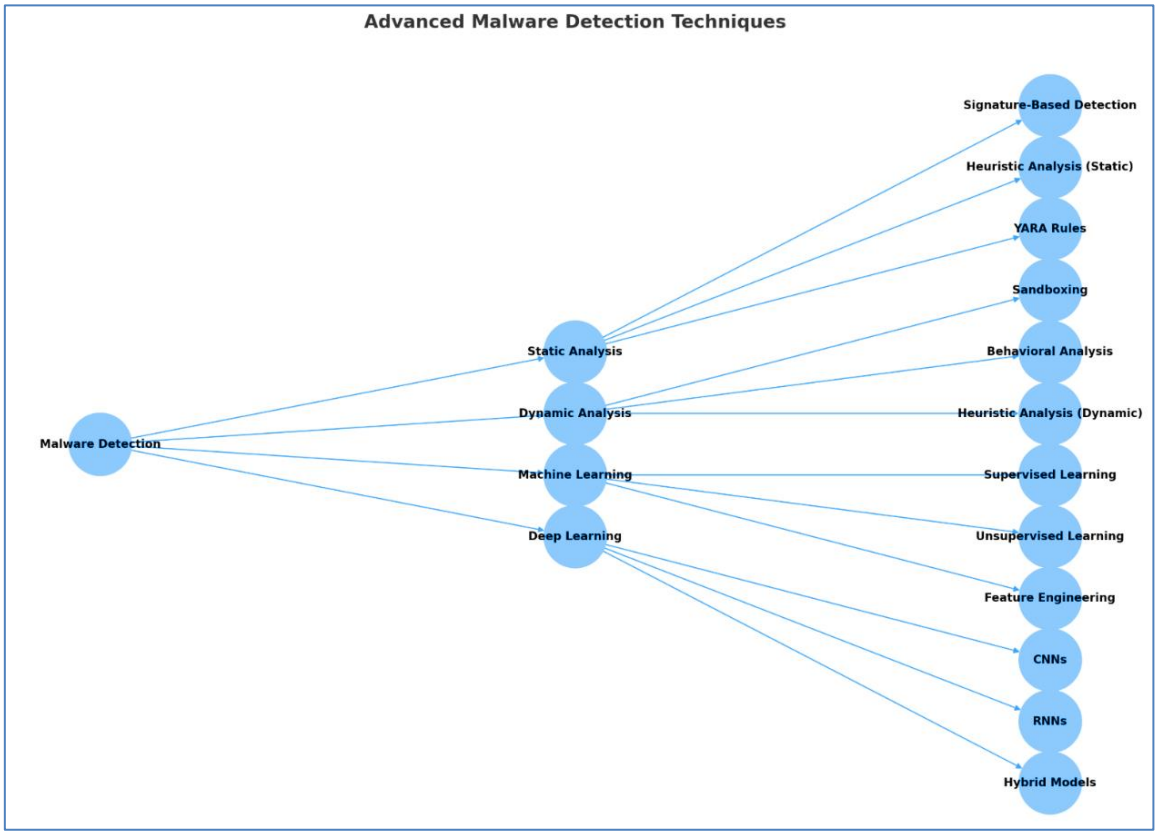

Fig.2. Advance malware detection techniques

4.1.1 Static Analysis Techniques

Based on how the malware samples are detected, there are two distinct approaches in malware detection and classification: static analysis techniques and dynamic analysis techniques. Static analysis techniques reverse engineers the malware samples. This process determines the structure of the malware binary code and obtains features to describe signatures of malware samples for specific families[22]. Researchers have attempted to collect attributes of malware samples such as code structure features, opcode sequence features, API calling features, graph s features, and so on. On the other hand, malware detection using static analysis techniques can be thwarted through the use of polymorphic and metamorphic malware as well as popularly used obfuscation techniques to pack and encrypt the malware binary code [23]. Dynamic analysis techniques, in contrast, involve executing the malware samples preferably in a sand-boxing technique in an execution monitored and controlled environment to model the behavior of malware. This step is important for understanding additional behavior of malware samples that cannot be captured through static analysis[24]. Additionally, malware samples can be de-obfuscated and unpacked through network traffic analysis, API tracing, memory forensics, and executed process analysis which are carried out via dynamic behavior analysis. Dynamic analysis requires a controlled environment to run malware code samples. Such techniques are computing intensive because most of the operating systems software, applications, libraries, kernel, file systems need to be installed in compliant manners similar to computing systems provided by targeted users. However, detection accuracy tends to be better than static analysis techniques because these approaches target malware family behaviors rather than patterns of certain sequences of code. Also, AI techniques for detecting network intrusion, which is based on Port number, packet size, protocol number, TCP flag, source IP, destination IP, and source bytes, can similarly be applied in static analysis and dynamic analysis [25].

4.1.2 Dynamic Analysis Techniques

Malware samples are analyzed usually through static and dynamic analysis. Static analysis involves the study of the internal structure of the sample and does not involve execution. This method is less compute intensive but tends to miss obfuscation techniques [26]. In dynamic analysis malware samples are executed and monitored in a controlled environment to understand the runtime behavior of the malware. This approach is computing intensive but tends to have higher accuracy in characterizing malware samples [27]. This analysis can be conducted either through the use of hardware emulation or the use of sandboxing. Hardware emulation is a processor level technique which models the behavior of an Instruction Set Architecture (ISA) behaviorally, registers, memory space, buses, and caches[28]. A sandbox is an environment that is specially created for the analysis of malware. During this analysis, the actions of the malware are monitored and captured. There are two main categories of sandboxes, agent-based and agent-less . In agent-based sandboard systems, an agent is installed inside the virtual machine. This agent captures the network traffic and can also control the execution of the malware. This technique is slower than the agent-less technique but can capture better information. In agent-less sandboxes, a network monitoring device is installed outside the virtual machine[29]. This device captures all the incoming and outgoing interactions and analyses them. This technique is faster than the agent-based technique. The approach is described in more detail in Section 1.2.5.1. Evaluation of the presented systems are also done using the malware samples to demonstrate how to perform benchmarks and comparisons of dynamic malware analysis systems. These comparisons can then build the foundation for future works that aim to develop more efficient systems [30].

4.1.3 Heuristic-based Techniques

Machine learning based malware detection systems are classified into two major categories: misuse (definition, signature) detection and anomaly detection. Anomaly detection systems learn to build a model of normal system behavior, then detect attacks based on deviations from the model, which can be either misuse or unknown attacks [31]. Malware is often associated with malware families that contain some common characteristics or defining features. For dynamic or behavior-based malware analysis, it is more desirable to characterize the behavior rather than the specific codes that may change over time. Therefore, a misuse or signature-based malware detection model is preferred by most vendors and is currently used by various antivirus products and network intrusion detection systems (NIDS) in operation [32].
Machine Learning in Misuse or Signature Detection involves five major steps. Firstly, the data need to be collected. Most of the techniques are network or host-based [33]. For network-based detection, a few well-known ports are chosen through which most of the data will be collected. Data are collected in real time, passed, and pre-processed using packet capturing tools. Secondly, data pre-processing is performed. Generally, the raw data need to be pre-processed before they are acceptable for learning patterns. Rule generation is the next step [34]. When the pre-processed data are ready, the learned patterns are mapped to rules. Rules can be either a sequence of characters to identify patterns of ASCII data packets (signature) or mathematical expressions to recognize behavioral patterns of non-textual data packets (misuse). Fourthly, learnt rules are utilized for denial of service (DoS) or other security response strategy alongside the misuse techniques. Denial of service (DoS) requires targeted sites or networks to be capable of enduring attacks [35].

Heuristic-based Techniques. The machine learning approach to malware detection focuses on generating rules based on the previous results of static analysis and/or dynamic analysis. Generated rules are then stored in an inspector engine in order to guide the inspection of the extracted data. Heuristic-based techniques can be interpreted as a combination of static and dynamic analysis. During the inspection stage, rules can be applied to method calls from the static analysis as well as API systems or the process control flow from the dynamic analysis [36].

4.1.4 Machine Learning Approaches

Artificial Intelligence (AI) pertains to a system or machine that imitates human behavior to respond in a smart way. AI is broadly divided into two categories: Narrow AI and General AI. Narrow AI is also known as Weak AI, which exemplifies a program that has been trained for a specific task, such as image classification [37]. General AI is also known as Strong AI, which represents a system that could outperform humans at nearly every cognitive task. AI relies on machine learning, which is a subfield of AI. Machine Learning (ML) uses historical data to develop algorithms that can predict future occurrences. ML techniques are categorized into supervised learning, unsupervised learning, semi-supervised / combination learning, and reinforcement learning. EL, text clustering, segmentation, and topic modeling are the dramatic tasks that are addressed by ML-based unsupervised learning recommender systems. The recommender system has various types: content filtering, collaborative filtering, demographic filtering, and knowledge-based. Beside AI, Intrusion detection is characterized by the identification of unwanted or unneeded messages into a computer. Internet intrusions, which are actions that threaten the confidentiality, availability, and integrity of data, are among the most serious and fast-growing threats in the present-day technological environment. Various types of attacks on networks, be it any comprising sensitive information, are perpetuated by hackers for financial gain, personal benefits or may just for breaching security concerns. As prescribed by [38], attacks aimed on networks can be classified into Denial-of-Service (DoS) Attacks and Penetration Attacks – a DoS attack is directed on the network with a desire to slow down or shut down the network services and a Penetration attack on a network is aimed at getting unauthorized access to a designated system. Detection of intrusion attacks on networks is an indispensable task for malware detection systems as intrusions can have devastating effects on a business. In the preceding years, great strides have been made regarding supervised machine learning techniques in protecting networks against malware attacks [39].

4.1.5 Deep Learning Approaches

The increase in the use of web applications has led to a higher number of attacks against different servers. Attack detection at the host level and application server can be modeled with pattern matching, but the increased complexity of web attacks makes modeling very difficult over TCP/IP. Clustering approaches attempt to detect ports, attacks and botnets over network traffic by extracting features on real data but this has limited scalability. Thus, anomalous detection on network behavior becomes very important and the attention is drawn towards the use of ML techniques on this area. Network traffic can be modeled with graph techniques and its relations encapsulated with neural network architectures [40]. There has been a general interest in applying more complex graph transformations but they either require characteristic domain knowledge or are limited to specific metrics. Deep learning holds interesting potential in the realm of Computer Networks due to its theoretical sound approach which leads to applicable solutions. Deep Learning also improves the way networks are modeled, allowing richer feature representation, being able to learn metrics suited to the problem and efficiently adapting hyper-parameter models from simpler topologies [41] .

The most common Deep Learning architectures are classified as DNN, ANNs, CNNs, DBNs, AE, RNNs, LSTM and their variants. These algorithms usually focus their optimization in reducing a distance between the output and the desired values to find their parameters. DNN architectures create large hierarchies of overlapping function approximators. The general idea is to stack several layers, where each layer contains multiple neurons and where each neuron is associated with hyper-parameters, the parameters of the model, including weight and bias. This basic structure is able to form big feed forward network models by allowing at every node a non-linear activation function followed by a linear transformation. Hence, with great flexibility of structure, DNNs are able to model any arbitrary transformation which relates inputs and outputs[42].

4.2. Techniques for Network Intrusion Detection

Intrusion detection is the process of identifying intrusions and potential violations of computer security policies and network security between networks. An intrusion detection system (IDS) is a software or hardware application that detects intrusions on a network or computer [43]. In order to detect intrusions, operating system activity and system configurations are monitored, and various types of malicious activities are analyzed. The activity that can be monitored by the IDS is the system log that is maintained in a log file. These logs include system boot logs and logs of connections with the superior, inferior and peer systems. Intrusions are analyzed by abnormal system behavior. Misuse detection is the second approach used to detect intrusions. The training of the abnormal and misuse detection is based on a set of rules for signs of system attacks, which are generated from knowledge on the intrusions. A new night university field where bio measurement signs

of the entrance to the building at night are defined has been created. There are three types of intrusions: computer- and network-based intrusions, mobile intrusions, and intrusions related to the use of illegal software. Computer- and network-based intrusions detect unauthorized access or attempts to access a network node [44]. These security devices reside on the local computer or are attached to the segment of the monitored network. These devices operate by periodically interpreting log files or by inspecting network packets. The detection procedures used in these devices can be divided into three principal categories: activity monitors that keep records of user activity on the host and that notify security services when predefined thresholds are met, signature-based detectors that inspect files and programs to decode known harmful patterns, and fingerprinting mechanisms that keep track of specific files and detect unauthorized copies and changes in them. These devices vary in terms of criteria used to determine suspicious activity and performance, targeted objects for inspection, and requirements for detected intrusions when they are found [45].

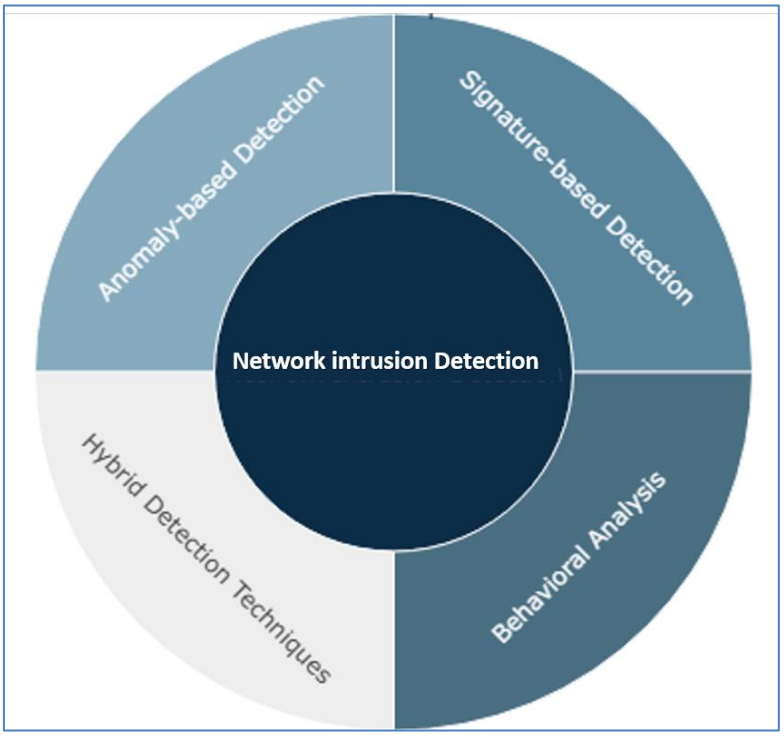

Fig.2. Techniques for Network Intrusion Detection

4.2.1 Signature-based Detection

Intrusion detection systems (IDS) play an important role in network security. There are various techniques and algorithms used to detect attacks on the network. These techniques are broadly classified into signature based detection, anomaly-based detection and hybrid detection. Signature based detection or misuse detection is a method that compares the observed behavior in the system or network with a known attack pattern. Detection is achieved by checking against the rules defined using a set of predetermined patterns or signatures. There are limitations on signature-based IDS techniques. New types of attacks may not be detected as they may not match with the signatures stored, hence signature based methods cannot detect novel attacks[46].

The signature-detection technique selects a technique for analysis based on the properties of the incoming packet. The preliminary info of the inbound packets is matched with the signature patterns. The parameters for matching processes are source and destination address, protocol number, source and destination port numbers and some more information depending upon requirements. If the condition satisfies the packet is considered an attack packet and the log details are stored in a file. The IDS will not inspect any further for such packets. On the other hand if the condition does not satisfy, it is a normal packet and it is passed to the next classifier or detection layer. The system only keeps track of the different attacks and there is no overheads of processing time. The technique identifies an attack quickly and it does not take time in processing. It is sensitive to detect some important attacks like: ZMOS, Chi-X, NEO, sgi0, rootshell, IPOD, GODOT [47].

The significant volume of personal info is out there. Most of individual information is transacted online and may well be misused by definite groups of people. Spam, phishing, malware and countless other varieties of attacks take advantage of

the vulnerabilities inside the network, hardware and OS. There is an paramount need to develop intelligent systems of intrusion detection for networks. Training an intelligent agent involves feeding it plenty of sample inputs, some of which may not correspond to any human labelled patterns.
A signature-based intrusion detection requests programming patterns for each possible pattern of network attack. The distinct protocols to be monitored were mentioned explicitly as a result of an IC mark format that permits for simpler written report, modulating and execution [48].

4.2.2 Anomaly-based Detection

Anomaly detection systems define normal states in a network and then detect system states that significantly differ from the normal states. After observing a network state pattern, the processing is focused on whether the pattern of the observed state differs significantly from the pattern of normal states. Any state that significantly differs from the normal state indicates a possible event of an attack [49]. Anomaly detection systems can detect new attacks on a network. A design challenge of anomaly detection systems is that if the normal state patterns do not significantly differ from those of the anomalous states, attacks will go undetected, and the false alarm rates will heavily increase. For the design of the anomaly detection systems, it is critical to design a normal state such that the patterns of the normal state maximize the detection rate while keeping the false alarms within an acceptable limit[50].
For a successful decision on the selection of misuse and anomaly detection systems for hybrid detection systems, the applications should be kept in mind. The way that an anomaly detection system is integrated with a misuse detection counterpart can be classified into the following four categories [11]. Anomaly-misuse sequence detection: In this case, misuse attacks are performed after first succeeding with an anomaly attack. The sequence of events takes place in time across two systems. Complex query generation and rule management mechanisms can be used to coexist. The anomaly detector can detect a subset of anomalous events before a misuse-style detection takes place. The architecture of the hybrid system can follow that of the misuse detector to some extent. This type of integration includes a plethora of options and variations for the design. Misuse-anomaly sequence detection: This case involves a scenario where misuse detection can take place before anomalous detection. If misuse-detecting rules can be generated based on the designs of the anomaly models, the integration will be similar to the previous category but focused on a different order of operations. However, this also calls for adequate care in managing the tradeoffs between false negatives and false positives, which can be intricate for more complex systems [51].

4.2.3 Hybrid Detection Techniques

Presents a hybrid intrusion detection system based on machine learning and shallow neural networks to detect and classify unknown intrusion actions. To deal with the exponential pain on incoming traffic volume, it employs a two-tiered architecture, where the first-tier filters out harmless traffic[52].

4.2.4 Behavioral Analysis

Applications operating under different platforms exhibit peculiar characteristics both at the static and dynamic levels of analysis. Factors such as the execution environment, the base API libraries, and supported platforms impose default behaviors on the applications that interfere with the predefined functionalities of these applications, hence known as the “normal” or “benign” behaviors . Deviation from the expected behaviors leads to suspicious applications that exhibit a behavioral imprint that characterizes their actions. The latter constitutes a dictatorship in the formation of a behavioral analysis approach. Each platform presents its proprietary analyzers, analysis methodologies, and types of analyzed applications. Behavioral detection systems on different platforms present similarities, diversifications, and uniqueness [53].
Feature Classification uses Machine Learning techniques to classify the characteristics of the analyzed application by referring to the generic rules and models previously built based on the characteristics already extracted from the application samples during the training phase of the system. Evaluation/Decision involves comparing the characteristics extracted from the analyzed application with the signatures or behavioral model previously stored in the system database to deduce whether the analyzed application is malicious or not [54].
MADAM classifies the behavioral characteristics of the known malware in six misbehavior classes. It ensures the extraction, analysis, and correlation of functionalities of the installed applications with the behavioral models to effectuate the detection and blocking of malicious activity. Experimental results have shown that MADAM is capable of detecting more than 96% of malware applications while keeping the low rate of false alarms when approving benign applications. M0droid comprises two modules: the light client agent and the analyzer server. Installed on mobile devices, the client agent communicates with an analysis server that analyzes the behavior of Android applications and creates signatures representing malicious activities. Minimizing resource consumption and phone energy is essential in mobile environments since performers run the majority of the Android applications[55].

4.3 Datasets for Malware and Intrusion Detection

The datasets used and generated for research in malware classification, and networks intrusion detection are detailed in this section. It is subdivided into three parts:

1) malware datasets
2) network intrusion detection datasets
3) dataset generation.

The following dataset is publicly available for research on malware detection. MISP Tools are part of the Malware Information Sharing Platform (MISP). MISP represents an open-source threat intelligence platform, where organizations can store, share, and correlate indicators of compromise and threat data. MISP data can be generated in a variety of formats, including CSV files, which are easier to manipulate and analyze than JSON files. The CSV files provided here are simpler and easier to use and further modify for specific scenarios. MISP stores indicators, events, and event objects using the Event and Object models. MISP Workbench is an analytic front-end that acts as a layer above the MISP back-end. It provides more advanced analytics capabilities for incidents with a higher level of specificity. Supported indicators, as well as the objects to be created and visualized, include Network Traffic, Domain Name, HTTP Request, IP Address, and URL. The latter is anonymized to obscure potentially sensitive data such as IP addresses and releases small groups of attributes. The anonymized files are easier to share across systems without compromising sensitive information[56].

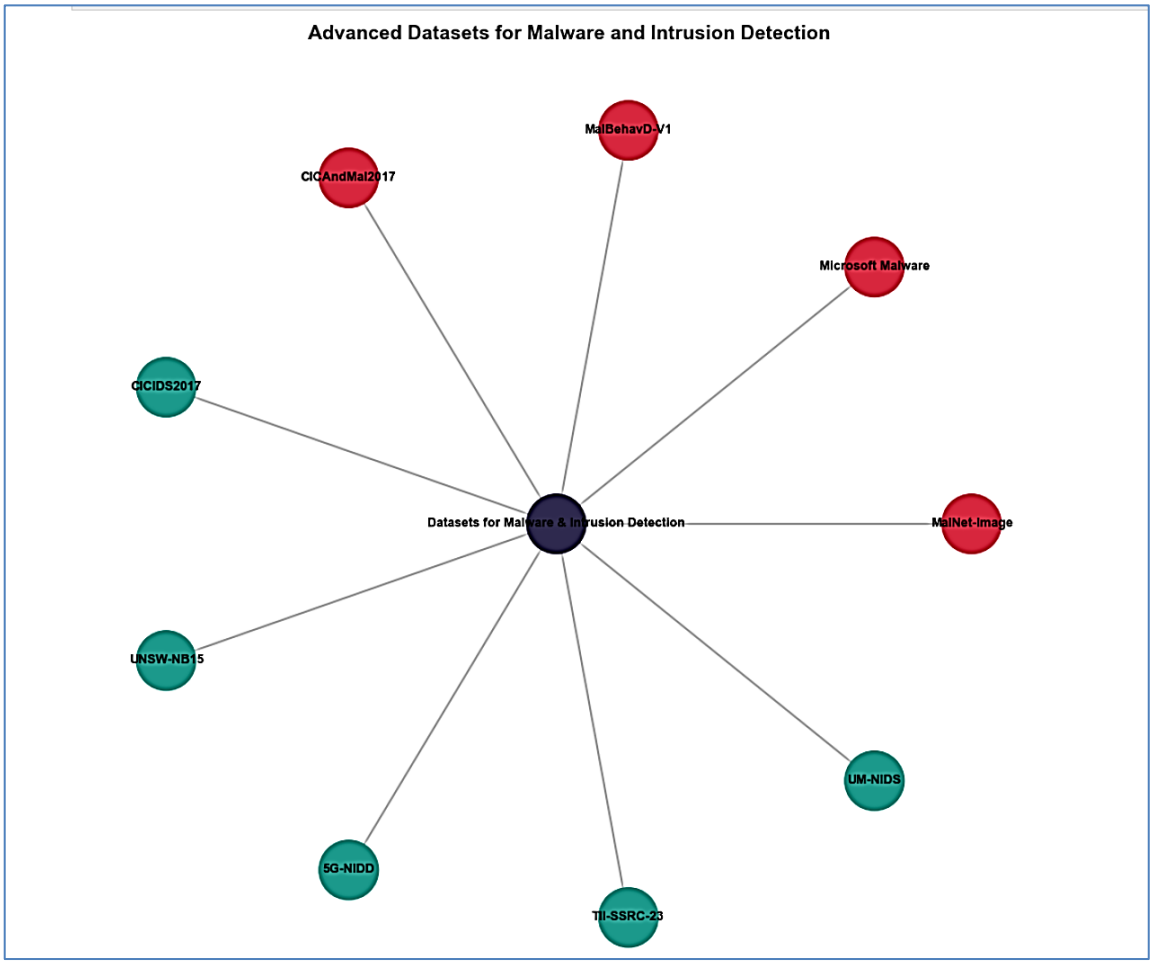

Fig.4. Datasets for Malware and Intrusion Detection

The datasets used and generated for network intrusion detection research are detailed in this section. It is subdivided into three parts: 1) network intrusion detection datasets, 2) record events and attack datasets, and 3) dataset generation. The datasets used and generated for both research areas are publicly available for easier reproduction, to help evade potential biases toward certain datasets, and to foster research in computation resources-limited environments. IDS datasets are made

available in a variety of formats and tools. For easier implementation and use, all datasets are represented in commonly used SQL databases and CSV files [57].

4.3.1 Publicly Available Datasets

Network-based datasets, developed by capturing traffic data in the real world, are the basis of all experiments in this work. These datasets are publicly available. A description of data collection protocols and a summary of data characteristics are provided below. Experimental results, including the accuracy and visualization of results on individual events, are also provided. Furthermore, the datasets used for training and testing automated detection models are available[58].

One dataset has been developed from a network with normal and anomalous traffic in a real-world environment. Labeling is conducted carefully to include traffic involved in botnet, DoS, flooding, packets, port scanning, exploitation tools, keys, and worms. Normal traffic is collected using common applications, such as browsing, file transfer, and video streaming. The entire dataset has been divided into a training set containing only normal traffic and a testing set containing both normal and anomalous traffic at different volumes. This dataset contains normal traffic and anomalous traffic events. Labeling is conducted manually to ensure the accuracy of the generated events, and no false events are labeled. The data was generated by the analysis of data capture. Flow data are obtained from files using tools, and text files are generated containing the two types of data[59].

In this work, an automated traffic event detection model is proposed, which contains three phases: (1) argument generation to preserve all possible behavior characteristics of traffic events from a small portion of the events; (2) argument labeling based on the KNN algorithm; and (3) argument detection based on the AdaBoost classifier. Other detectors such as XGBoost and LightGBM can be easily integrated into the detection phase. Coping with small labeled data has always been a challenge in the field of automated detection. To deal with this issue, the focus should be moved towards preserving all possible behavior characteristics of traffic events from a small portion of the labeled events rather than providing a few key characteristics. Such work is normally not performed in most of the existing studies. A mechanism is proposed to generate arguments that consider both the behavior and syntactical characteristics of traffic events. The generated arguments can preserve the properties of every traffic event with the corresponding ID, including both normal and anomalous behavior. This is a solution that is pertinent for general settings, general languages, and general descriptions[60].

4.3.2 Synthetic Datasets

Nowadays, cybersecurity plays a fundamental role to ensure the usability and integrity of the information technology and telecommunication infrastructure. For such reason, Network Intrusion Detection Systems (NIDS) and Intrusion Prevention Systems (IPS) had been incorporated machine learning (ML) and deep learning models to detect malicious network traffic patterns with excellent results. The implementation of a NIDS is currently one of the actions that organizations have taken to avoid data and service breaches. Such systems are used to monitor and analyze network traffic, and alert or take protective actions in case of malicious attacks and activities. As the rate of producing data grows, NIDS has become indispensable and increasingly relevant. However, there are several research challenges and public concerns involved, such as relevant features selection, proper models to evaluate, and large and representative datasets to train and validate the models. To accomplish desirable features on a NIDS, it is necessary a large and comprehensive experimentation and modeling work, where most of the available conditions of work for a NIDS be considered. Usually, those conditions are represented through datasets[61].

Over time, several public network datasets had emerged, these often have their heyday and are subsequently being displaced by updated versions or new datasets with better features. Automatically generated datasets were produced from simulation of network scenarios that captured normal activity and a limited number of malicious actions. Only recently, Managers of successful cyber security companies produced dynamic and synthetic datasets with a specific profile[62]. Such datasets allowed the creation of network scenarios with many categories of attacks, as well as continuous traffic that leads to more realistic behavior of malicious actions. In those scenarios, in addition to monitoring the NIDS operating in real time, log files were recorded to assess malicious traffic as well. The main goal of this research is to compile and implement a synthetic network traffic generator, and analyze the strengths and weaknesses of the two widely used architectures: HMM and LSTM-based generator to automatically produce network traffic logs. In addition, a solution to normalize and anonymize traffic logs and translate them to CIDDS format is shared as open-source to assist researchers to generate logs for NIDS simulation through simulation [63].

4.3.3 Challenges in Dataset Construction

The implementation of artificial intelligence in any area depends highly on a large number of data points for training, learning correct behaviors and patterns, and ensuring categorization and ranking. AI implementation in malware and network intrusion detection is no different. It requires a well-structured and informative dataset for its implementation. Dataset construction for AI applications, while important, has a number of challenges. They include dataset representation, dataset creation, the class imbalance problem, and label noise[64].

There is an increasing tendency toward artificial intelligence-based methods to outperform traditional methods in data representation because feature extraction processes are adapted to the nature of the input. However, many feature learning tasks require hand-crafted mappings, rendering mX-representation approaches inherently inferior for many AI systems and leading to the proliferation of new efforts. Consequently, it is inadequate to express knowledge in a format that is appropriate for subsequent analysis. Moreover, the intensity and volume of data to be processed would increase tremendously over time, which would impose a burden on storage, transport, and analysis complexity [65].

Dataset creation generally refers to either collecting information/data from existing databases or designing one's own specific datasets. Dataset creation for practically usable messages on the web is challenging because of spam filtering. The main problems with web-based datasets are daily updates, the demand for online trawling systems, and the robustness of the filtering routines. Practical datasets for malware classification, which comprises benign and malicious executables, are sensitive as well; for the former, collecting such codes in a crowd-sourced way is plausible, though not easy. In contrast, the collection of malware samples in an automated manner is quite complicated [66].

5. COMPARATIVE ANALYSIS OF TECHNIQUES

A comparative analysis of the various techniques adopts a multi-fold approach, where the techniques are categorized based on their distinct characteristics, and then compared to identify the advantages and disadvantages of each technique. The categories chosen include (1) based on learning mechanism employed for classification and detection, (2) based on features used for training and detection, (3) AI techniques employed, and (4) based on the deployment.

TABLE I. A COMPARATIVE ANALYSIS OF THE VARIOUS TECHNIQUES ADOPTS A MULTI-FOLD APPROACH

Category	Technique / Study	Features Used	Advantages	Weaknesses / Limitations
1. Learning Mechanism	Supervised Learning (e.g., Decision Trees, SVM, etc.)	Static features, labeled datasets	High accuracy in controlled settings	Poor generalization with small or imbalanced datasets
	Unsupervised Learning (e.g., Clustering, Autoencoders)	Behavioral logs, network patterns	Detects novel threats, no labeled data required	High false positive rate; less interpretability
	Reinforcement Learning	Action-reward feedback from detection environment	Adaptive to changing threats	Complex and resource-intensive
2. Features Used	Static Analysis (used in most reviewed techniques)	Byte sequences, opcode n-grams	Fast, low resource usage	Easily evaded by obfuscation or packing
	Dynamic Analysis	API calls, system behavior, logs	Captures runtime behavior, resistant to evasion	Requires sandboxing; high overhead
	Hybrid (Static + Dynamic)	Combination of above	More robust, improved accuracy	Computational cost; harder to deploy
3. AI Techniques Employed	Deep Learning (CNN, LSTM, etc.)	Raw binary, logs, sequences	High detection accuracy, automatic feature learning	Needs large datasets, risk of overfitting
	NLP-Based (Document structure, macro analysis)	Word structure, macros in documents	Innovative, useful for document-based malware	High computational cost, limited generalizability
	Ensemble Methods	Combined classifiers (e.g., RF + DL)	Improved robustness and performance	Complexity in integration and tuning
4. Deployment	Standalone Applications	Offline analysis	Easy to implement and test	Not suitable for real-time or mobile use
	On-device / Edge-based	Localized model execution	Enhances privacy, enables real-time detection	Limited by device resources
	Cloud-based / Hybrid	Cloud for heavy processing	Scalable, centralized learning	Latency, privacy concerns

The table above presents a comprehensive comparative analysis of AI-driven approaches to malware and network intrusion detection that are categorized by learning mechanisms, feature types, AI methods, and deployment options. Each class is evaluated based on its detection accuracy, computational intensity, adaptability, and suitability for real-world implementation. Supervised machine learning techniques have high accuracy in labeled settings. Unsupervised and reinforcement learning provide flexibility and new threat detection but are limited by complexity and interpretability. Algorithms based on features like dynamic, static and hybrid analysis sacrifice efficiency for resilience, with hybrid models offering improved accuracy at additional computation. Of AI techniques, deep learning and ensemble approaches offer improved performance at the expense of big data and complex integration. Deployment models vary in scalability and latency, with cloud solutions having centralized management and on-device solutions offering privacy but hardware

limitations. The findings highlight the importance of hybrid and context-aware solutions to balance performance, cost, and deployment practicability in today's cybersecurity systems.

6. CASE STUDIES

With the ever-increasing threat landscape, the importance of applying AI technologies to malware detection is becoming increasingly apparent, particularly on the Android platform. Importantly, the founding literature, notable limitations in these works, and potential areas for future research are discussed. Given that Android is the most popular mobile operating system in the world, there are inherent security concerns. Furthermore, malicious applications have proliferated on mobile platforms, leading to increased needs for new, different methods to detect. As a result, Android malware detection is challenging and has become the research focus of many academicians and practitioners around the world [68].

Based on the analysis of Android applications (i.e., APK files), many methods for detecting malware have been proposed. Unless indicated to the contrary, most of the more recent works discussed here focus solely or primarily on static or permission-based detection approaches. It has been pointed out that static detection methods, specifically those based on permission analysis, are limited in scope since not all malicious applications use dangerous permissions. A machine learning model—the RF-API permissions (API request of applications)—to indicate the maliciousness of an unknown application, achieving great performance. However, the distribution of permissions has been imbalanced in both datasets, potentially misleading the model results[69].

A requirement on the enablement of permission identification may limit the commercial application of their work. In addition, a specific dataset may not be representative enough to indicate the overfitting/underfitting of the model. Many prior works reported experimental results solely using older datasets, which should be augmented with other, newer Android datasets. In addition, the latest, more state-of-the-art machine learning models, such as zero-shot learning models, attention mechanism-inspired models, and knowledge graph-based models, should be explored to ensure a lower false positive rate. Meanwhile, given the inherent dynamic characteristics of Android applications, many works have applied dynamic analysis to detect Android malware. These works generally leverage sophisticated runtime environments to monitor the behavior of suspicious applications and identify security violations. However, they are still challenged by the inherent complexity of behavior analysis, inaccessible runtime logs, compute-intensive judgment mechanisms, and other limitations on classifying structures[70].

6.1. Successful Implementations

Accurately predicting long-term network traffic patterns is critical, as malicious nodes and gateway nodes tend to remain active over extended periods in real-world scenarios. Much of the current research that focuses on statistical traffic prediction relies on PCAP files and conventional classifiers, which are often too slow for real-time inference, making them impractical for deployment in live environments [71].

Many existing traffic prediction methods do not pay sufficient attention to multi-step forecasting, even though predicting long-term traffic behavior is more valuable than simply forecasting the next step. Models based on Recurrent Neural Networks (RNNs) and their variants typically focus on single-step predictions and lack interpretable architectures, which is a critical requirement in sensitive systems that demand explain ability [72].

So far, most defense mechanisms concentrate on binary traffic classification and fail to consider threats like benign malware-injected applications (MIAs) or predicting the threshold of Denial-of-Service (DoS) attacks (Choi et al., 2020). Furthermore, approaches based on Artificial Neural Networks (ANNs) often lack proper explanations of the input features, limiting their applicability in environments requiring transparency and accountability [73].

A rule-based tool that can simultaneously detect malware and penetration testing tools represents a promising advancement. It is among the first to offer clear explanations for its predictions. However, it relies on a limited-capability commercial sandbox and cannot be openly distributed due to usage restrictions, which hampers comparison and widespread adoption [74].

Other architectures based on deep learning that consider previous context—although rarely used in this domain—are often too complex to be understood by human reviewers. Despite their power, their high computational cost limits their effectiveness for real-time applications [75]. Some in-house approaches rely on a limited set of lexical and syntactic features to evaluate targets. While these methods are more interpretable, they often fail to provide comprehensive coverage of newer, more promising features, resulting in incomplete detection due to the narrow feature scope [76]. Finally, a class of solutions focuses on identifying unknown executables by analyzing opcodes or other static methods. However, these approaches often lack robust feature vector generation mechanisms necessary for accurate and scalable malware detection [77].

6.2. Lessons Learned

Malware encompasses various types of malicious software designed to infect, disrupt, manipulate, or damage computers, networks, and connected systems. The rapid proliferation of malware strains in recent years reflects the growing

professionalization of cybercriminal activity. Modern information systems are constantly exposed to threats that may compromise their confidentiality, integrity, and availability[78].
Traditionally, perimeter defenses and antivirus software installed on endpoints have served as the primary lines of defense. However, to combat persistent and targeted threats, early-stage detection is essential—ideally during normal system operation and data flow phases [79]. Machine learning (ML) has emerged as a powerful tool in this context, enabling the analysis of vast amounts of data to identify patterns indicative of malicious activity media.kaspersky.com. Malware detection is commonly formulated as a binary classification problem: identifying whether executable files are benign or malicious [80]. Since benign files vastly outnumber malicious ones, this imbalance introduces challenges such as high false positive rates. ML algorithms have been successfully applied to address this issue, with studies demonstrating high accuracy in detecting obfuscated malware variants ScienceDirect. However, to be viable in real-world environments, these models must not only perform well but also adapt to evolving threats, ensuring continuous learning and avoiding performance degradation over time par.nsf.gov[81]. Another critical consideration is model interpretability. Analysts must be able to understand the rationale behind a model's decision to build trust and support further investigation. Recent research has focused on enhancing the interpretability of ML models in malware detection, employing techniques such as decision lists and Shapley additive explanations (SHAP) to provide transparent and explainable results[82] .

7. CHALLENGES IN AI FOR CYBERSECURITY

In recent years, AI has experienced significant advancements, and their integration into cybersecurity systems can greatly enhance organizations' data protection. By leveraging the latest methodologies and technologies, highly efficient detection, prevention, and reaction systems can be created against a wide variety of cyberattacks. To achieve such an outcome, the entire architecture must be appropriately designed and deployed, with the AI-enabled tools playing a crucial role once all data is acquired. One of the primary advantages of AI in cybersecurity systems is the volume of data it can cover, allowing organizations to examine large amounts of logs and analyze a wealth of information for intrusion detection. Additionally, AI-based tools can identify patterns and recognize anomalies in data analysis. Utilizing a segmented network with a decentralized architecture adds robustness to the system, making it less prone to attack and performance degradation [83].
Some considerations must be addressed during the preparation and deployment of an AI-enabled cyber defense system. To offer a relevant solution against potential attacks, the underlying data infrastructure must be capable of dealing with large volumes of data. Currently, organizations generate, collect, and store extensive logs, often across different departments and services. A cloud-based solution on which the data would be centralized would be beneficial, as well as distributed processing to cover the high throughput in terms of the amount of logs produced. Once the data infrastructure is set up, the AI algorithms must be integrated with the existing overall IT architecture, with particular attention given to the plug-in point and input/output streams to analyze data in a timely fashion [84].
The ML pipeline must be designed to operate autonomously, reducing false positives and false negatives. To achieve such an outcome, rigorous validation of the data must be guaranteed, enabling the detection of various kinds of anomalies. Since data inevitably contains noise and outliers, they must be detected and removed as soon as possible[85]. Data bias is another critical consideration in AI approaches, whereby the training dataset influences model performance. Creating an unbiased representation of reality is impossible, as it can only approximate it. Therefore, efforts must be made to reduce bias during training data generation, which must be updated periodically[86].

7.1. Data Privacy Concerns

Learning, which is the process of extracting knowledge from raw data, occurs at different levels. There are two basic forms of learning: supervised and unsupervised. Supervised learning requires examples of the desired output along with the corresponding features to induce the mapping[87]. In unsupervised learning, datasets aren’t labeled and instead consist solely of the raw data. The learning algorithm is required to employ the input instances to discover structure within the data. For example, unsupervised learning can extract groups of samples that are similar to one another based on metrics associated with the features describing the samples[88].
In supervised learning, once adequately trained, the resultant model is able to predict a label for a test sample (one that has only the features describing it, and is not included in the training dataset. The key is that no information about samples in the training dataset, other than what can be inferred from the trained model, should be exposed. In both forms of learning, there exists a chance that an adversary can learn unintended knowledge about the training samples. Machine learning practitioners should not have to worry about information leakage associated with machine learning systems[89]. Privacy in machine learning is concerned with exploiting sensitive private data inappropriately. For example, the disclosed labels of supervised training instances can be exploited to carry out model inversion attacks, membership inference attacks, and data reconstruction attacks. In the field of unsupervised learning, adversaries can exploit auxiliary knowledge to discover training instances whose features are close to them and the features of a predictive learning model can be developed to help the adversary in this task[90].

7.2. Adversarial Attacks

In recent years, machine learning (ML) has emerged as a new solution to the dilemma of the increasing sophistication of threat landscapes while not having enough qualified skill professionals in cybersecurity . It brings great hope to improve the detection rate and efficiency of IDS. However, as a double-edged sword, the use of ML for security also raises new questions in the form of adversarial attacks. Many studies in the field of computer vision/image classification have illustrated that ML model can be fooled by adversarial attacks. These findings sparked a new wave of research on the feasibility of adversarial attacks against ML-based security systems, such as spam filtering, firewall, IDS, etc. Most of the existing studies have shown that it is possible to mislead a model with adversarial examples created using features derived from a raw data source like images or packet capture [91]. Despite the findings that a plethora of feasible adversarial transformations exist against ML systems such as firewalls or spam filters, the feasibility of craftable adversarial examples has never been shown in the context of NIDS that operate on the network layer. This study investigates the actual feasibility of adversarial evasion attacks against network-based intrusion detection systems (NIDS). The current work demonstrates that it is possible to fool ML-based IDSs with its proposed adversarial algorithm in a black-box setting. As defense, a generic defensive scheme is realized to detect and protect ML-based IDSs against adversarial evasion attacks, and this work illuminates new research opportunities in adversarial attack and defense for ML-based NIDS as well as general ML-based network security systems[92]. With the subject being examined as IP packets and two NIDS datasets selected for the experiments, realistic botnet traffic traces are effectively created and used to assess this work. This work aims at studying the deed problems of adversarial botnet traffic generation that avoids detection while still performing its malicious functionality. First, under the generative setting, a framework named Adv-Bot is proposed to create adversarial botnet traffic. The main process relies on a learning-to-generate mechanism where two agent networks are jointly trained to counteract against one another and evolve until a suboptimal strategy of traffic generation can be learned by the generator. Second, extensive experiments are conducted for thorough evaluation through various datasets, botnets, and NIDSs, and the effectiveness and generalizability are verified. Third, realistic adversarial samples are delivered to the public for future research on both evasive attacks and defenses against ML-based NIDSs[93].

7.3. Scalability Issues

One of the major scalability issues with a system based on a static set of rules is the linear growth in computation after each rule is added. Every possible protocol or keyword should have its own set of rules for which packets should be checked, and every new rule should be carefully placed in each interval for every protocol. The exponential growth of the decision tree leads to either an overflow of the hardware implementing it or a processing speed that is too slow to be feasible, especially as the constant increase in processing speed and complexity necessary to keep up with the growth of networks, as networking tubes grow larger and larger in size[94].

Perhaps a better solution to the difficult problems of scalability and complexity, as well as the impenetrability and unworkability of many artificial intelligence schemes, is a simpler detection algorithm that produces a more human-readable output without the need for large-scale algorithmic representation. One additional advantage of simpler methods is that they can be summarized quickly, and the examination of only a fraction of the packets will show a good characterization of the whole input stream. Results show that simpler methods are both more efficient in finding the most crucial attributes and easier to implement in practice [95].

Even simple methods can benefit from additional dimensionality reduction after construction — for example, the technique of discretizing wavelet coefficients via hierarchies can be applied. Simple methods accept fewer or lower level inputs and therefore can respond faster both in actual data searched and on the individual data. Since for many simpler methods the absence of the addition is equivalent to the absence of the individual attribute, they are usually much faster than either sum or product measures. Moreover, a large body of so-called unsupervised methods does not require a hand-constructed database and can classify raw signal since their results depend on the separation boundaries of the set of training signals, independent of the internal details of the learning algorithm [96].

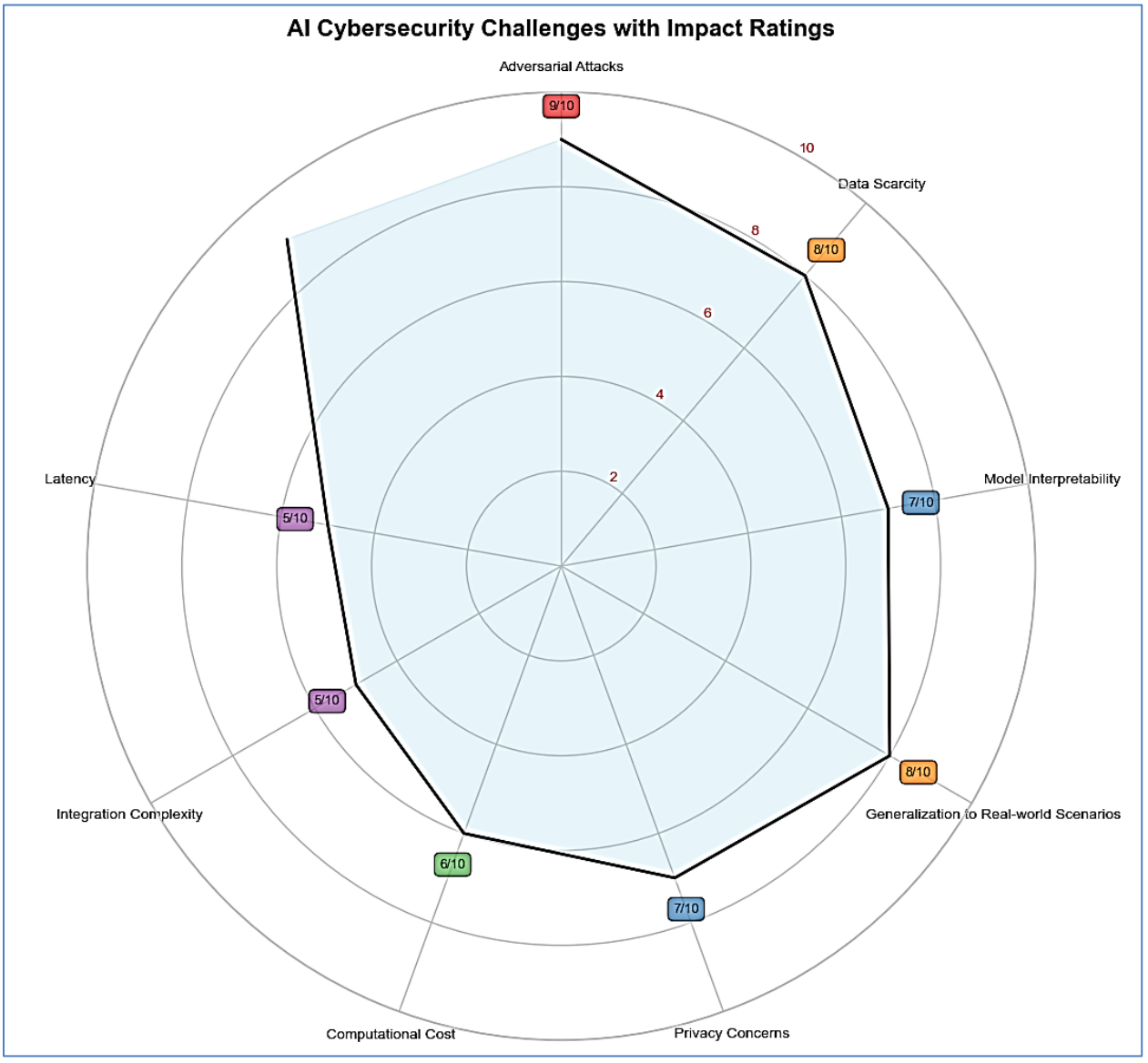

Fig. 5. AI Cybersecurity Challenges with Impact Ratings

From figure (5)The radar chart shows the impact of various challenges in AI cybersecurity, with challenges such as Adversarial Attacks (9/10) and Data Scarcity (8/10) being the most influential, indicating their research and practical priority levels. Other challenges such as Model Interpretability (7/10) and Privacy Concerns (7/10) are also significant, while Integration Complexity (5/10) and Latency (5/10) are less significant. Such ratings reflect the level of urgency and frequency of discussion in the field, prioritizing the research and solution for addressing the most pressing setbacks in cybersecurity due to AI.

8. FUTURE DIRCTIONS AND RESEARCH PERSPECTIVES

Although the research has proven the efficacy of the proposed network intrusion detection and malware detection methodologies whose performance is comparable to that of the state-of-the-art techniques, there exist further avenues of exploration in the realms of network intrusion, malware detection, and better utilization of AI tools. A collection of possible future explorations is put forth in this section, while other future perspectives that had better remain unexplored are also listed to guide from side-tracking.

8.1 Improved Detection Methodologies

Improved Methodologies. More advanced detection methodologies may be developed by employing a collection of techniques that have not been utilized in this dissertation. For example, decision tree-based models, random forests, and

Monte Carlo tree search techniques have great potentials for modeling and detecting both network intrusion and malware. An in-depth exploration of Unscented Kalman filter (UKF) may yield new insights for network intrusion detection. More real-time and robust solutions to some of the attacks with known categories may be developed by leveraging swarm intelligence-based algorithms such as swarm-based detection.

8.2 Utilization of Emerging AI Tools

Utilization of AI Tools. New AI tools with great potential may be introduced for R&D in malware detection and/or network intrusion detection. For example, AutoML tools may have the potential for improving R&D procedures. It may be interesting to explore whether they can successfully develop better detection methodologies than the ones developed in this dissertation.

8.3 Next-Generation Mobile Malware Detection

Next-Generation Mobile Malware Detection. While accepting that multi-faceted smart devices have brought conveniences to people's daily life, it has also raised new cybersecurity issues including but not limited to privacy breaches, telemetry attacks, compromise of rail systems, and adversarial attacks against defense models. Effective malware detection methodologies with at least single-character detection granularity trained on datasets incorporating the scenarios above need to be developed.

8.4 Research Areas with Limited Prospects

While the aforementioned topics may be promising for exploration, some other areas are considered to lack aspects for extended R&D in the current research topic. Active cyber deception, cyber forensics, and adversarial learning techniques may not be straightforward or have the potential to be competitive enough. Hence, they are not considered to deliver convincing outputs for the related explorations. Handling false positives and simulating attack datasets for mobile systems may be harder to achieve easily understandable perturbation and therefore less appealing options.

8.5 Emerging Technologies and Their Cybersecurity Implications

Emerging technologies like Artificial Intelligence (AI), the Internet of Things, and Blockchain are all set to drastically reshape their respective industries. However, alongside these advancements, concerns around cybercrime are rising as well. In recent years, Artificial Intelligence (AI) technologies such as malware and network intrusion prevention tools have been developed to counter the growing threat of cybercrime caused by the emergence of novel technologies. Malware detection using AI techniques like Deep Neural Networks, and Random Forest is highly efficient compared to traditional techniques. Enhancements in Natural Language Processing (NLP) have enabled the detection of fake reviews on social media platforms. Blockchain technologies, predominantly used in cryptocurrencies, are now being employed to create tamper-proof digital IDs and combat fraud in financial transactions [1].

AI-based Malware Detection is a major research area in the cybersecurity domain. The rapid growth of mobile networks has also contributed to an increase in the number of mobile malware threats, causing major losses to stakeholders. Malware detection techniques using Ant Colony Optimization, Decision Trees, and Deep Learning model classifiers based on Convoluted Neural Networks are explored. AI solutions in this domain are highly scalable and resilient against data drift and changing modes of attack. Anomaly Detection is another research area where the early detection of intrusions and threats to a computer system is remedied with AI techniques like Anomaly Detection Integrated Mac and Windows Logs Using a Deep Neural Network, Detecting Drone-based Security Attacks at Airports Using Machine Learning Techniques, and Reinforcement Learning for Attacking and Defending Networks.

8.6 Potential Research Directions and Dataset Engineering

The focus of this Survey is on the use of AI methods for malware detection in the mobile domain. The mobile malware ecosystem and available malware datasets were invoked for this purpose. By analysing the literature, eight broad groups of AI methods tailored for the mobile domain were identified. The discussed literature incorporates detection techniques that succeed in isolating malicious apps from non-malicious ones. Improvements that can be made upon them were identified, together with their applicability in the mobile domain. AI-based techniques for network intrusion prevention were also discussed. However, the scope with regard to the literature survey is considered complete since the most relevant papers were collected and examined. Many studies consider selecting pertinent attributes to improve detection results. Several attributes that help isolate malicious network packets from non-malicious ones were identified, and a small set of them was subjected to different selection methods.

The majority of the cited works concern implementing some classifiers that have proved successful across a broad range of datasets. Nevertheless, several classes of attention-based architectures have scarcely been applied. The network intrusion prevention community would benefit from conducting reproducibility analysis and reporting outcome variance metrics. New AI techniques can be developed or existing ones tweaked with the aim of improving performance across attributes that convey similar information. Here the attention-to-Attack Graph model is highlighted. This is a visualization tool that allows

the graph of steps an attacker can take to compromise a specific framed target to be generated. Finally, it is suggested that work should be done on creating a dedicated dataset catalogue with comprehensive annotations.
Referring again to the mobile security domain, a recent survey indicates that an up-to-date malware data source does not exist. New mobile datasets, whether labelled or not, are needed to immerse researchers in this domain. The work covered here will start with the most relevant datasets and describe why others were set aside. Automatically generating datasets from the mobile application store is also a compelling future direction. Currently available automated approaches would benefit from being trained via user feedback in order to generate tighter and better focused datasets. Furthermore, speculation is cast on how ML methods can be employed for this purpose. AI methods can be developed to balance datasets as this would greatly benefit many classification techniques.

9. ETHICAL CONSIDERATIONS

Today, with the increasing use of artificial intelligence (AI) technologies in combating malware, new ethical and technical challenges have emerged that require study and analysis. This section addresses three main topics: AI-powered malware, bias in training data, and accountability in sensitive systems such as robots and autonomous systems.

9.1 AI-Powered Malware and Reverse Engineering

Artificial intelligence (AI) has been broadly utilized as a tool to defeat the malware. This also brought a new problem: AI-based malware. This new type of malware was utilizing the advantage of AI, which stays undetected against the detection tools for a long period. Hence, reversing AI-based malware or malware's neural model is very important to make a specific detection tool for AI-based malware [2]. Kernel-based on reverse engineering, one of the AI models, is taken to show the process and challenges in reversing the malware model. The malware model container is restored by uncovering the defenses gradually, including dodging, monitoring, and characterizing attacks. The analysis system is threatened by the sample generator; thus, more research is required on obtaining the malware structural model or parameter [20].

9.2 Dataset Bias and Fairness in Machine Learning

The general concept of a bias is a preference for one hypothesis over another. In a more narrow sense, bias denotes prejudice by morally incomplete data if the application is in fields where data or specifications are generated, such as training datasets on which ML algorithms/models are trained [23]. The need for unbiased data or specifications is strongly motivated by three reasons. Firstly, and most critically, dishonest or biased assumptions can lead to major damage with catastrophic consequences. In the worst case, a wrongly specified job offering can even cause deaths or severe injuries. Secondly, even superior algorithms can return incorrect and unusable results if trained with biased datasets. Quality means proper lab our and solved assignment on the one hand. Quality also means the unbiased nature of the training datasets and unbiased specifications on the other hand. Lastly, accountability, legal or not, demands for a sort of explanation of why systems behaved in a certain manner. Luckily, most of the potential biases can be stated and recognized beforehand, i.e. before the generation of the training datasets takes place.
Bias, in general, refers to data specifications either too incomplete or too restrictive, necessitating a further refinement. This means that potential bias has to be defined in a way such that it can be measured. Therefore, it can be regarded as the more general term. In a more narrow sense, bias usually means prejudice by morally incomplete data, where the application is in training ML algorithms/models with data sets [24]. In this manner, bias refers rather to a data peculiarity and therefore to the specification of a real-word process that needs to be captured by other means. The bias is therefore more a data issue than a model issue and has to do with the quality of either the input/output data pairs or the input data. Potential biases need to be recognized and defined before training data sets are generated. This task is addressed in this section.

9.3 Accountability in Safety-Critical AI Systems

Ensuring accountability in AI systems is increasingly being recognised as a necessary part of trustworthiness. Accountability is broadly defined as the responsibility for expected or actual actions being conveyed to affected parties, whether through oversight, audit, enforcement, punishment or restitution of damage. Therefore, accountability is a prerequisite for trustworthiness. The widely-cited accountability framework proposed by Bovens' principles can be generally used to assess AI accountability, which includes the principles of answerability, openness, correctness and enforceability. In this paper, specifically developed accountability mechanisms, tools, standards and benchmarks applicable to AI systems in self-driving and robotics contexts are reviewed. Some are additional mechanisms aiming to enhance AI accountability, whereas others are tools intended to ensure that the principled AI systems are accountable or compliant. Compared with existing review papers, this review focuses on AI systems in safety-critical domains, i.e. robotics and autonomous systems, and offers a comprehensive classification of accountability mechanisms (P1–4) based on two criteria: (1) targeted principle in Bovens' accountability framework and (2) type of assurance provision (direct or indirect). Due to rapid developments of AI technologies and their products, recent AI accountability tools in safety-critical sectors have experienced a surge of interest and involvement from the parties including academics, industries, and policymakers, highlighting the growing demand for

accountability assurances. However, these mechanisms and tools are promising but remain immature, with significant extra development required before they might mitigate meaningfully the concerns of civil society [25]. Existing benchmarking practices for AI and software verification do not specifically address how AI-specific components and pipeline can be covered. Beyond the known verifiability problems of AI, little understanding on accountable AI tools ideas are captured in testing and benchmarks. New evaluators for written-knowledge provenance from general textual databases are being established, especially on large language models (LLMs), but methods grounded in cross-model data validation are still lacking and metrics are not internationally established [26]. More generally, how accountability and audit trails historically rely on clearly specified recording contrasts extremely with the ML "black-box" bottleneck where no intelligible representation of knowledge grounding and provenance can be modelled. An applicable metric-based standard for quantitative assurance records with interpretability will be developed.

10. CONCLUSION

AI-based malware detection techniques have garnered significant research attention in recent years. Numerous surveys have systematically categorized these techniques into multiple classes based on algorithmic characteristics and functional application. Prominent AI methods such as decision trees, Bayesian networks, deep learning, fuzzy logic, support vector machines, and genetic algorithms have been applied to malware detection, each offering unique strengths in handling specific threat models. In addition, prior studies have reviewed associated tools, datasets, programming languages, and distribution methods, offering a comprehensive view of the current landscape and its ongoing challenges.

In response to the emergence of more sophisticated malware, next-generation AI-based detection approaches have also evolved. To this extent, our study proposes a hybrid AI Malware Detection and Prevention Framework combining AI-based malware classifiers and network intrusion prevention systems (NIPS). The suggested hybrid framework seeks to improve real-time threat detection and prevention.

Furthermore, we discuss a number of baseline AI and deep learning architectures and introduce enhanced architectures for advanced malware behavior. The system is evaluated in real-world deployment environments and is demonstrated to work effectively on a wide range of advanced malware attacks. Our research also offers learnings on the financial, engineering, and data-related aspects of deploying these kinds of systems, paving the way for scalable and viable cybersecurity solutions. Future work can be extended to include explainable AI (XAI) techniques to enhance the interpretability and trustworthiness of malware detection systems in dynamic and adversarial environments.

Conflicts Of Interest

The author's affiliations, financial relationships, or personal interests do not present any conflicts in the research.

Acknowledgment

The authors extend appreciation to the institution for their unwavering and Special thanks to University of Kerbala, , Al-Ameed University and *Warith Al-Anbiyaa University* College Dean's University support and encouragement during the course of this research.

References

[1] R. C. Pivetta de Araujo *et al.*, "Serial mass screening for tuberculosis among incarcerated persons in Brazil," *Clinical Infectious Diseases,* vol. 78, no. 6, pp. 1669-1676, 2024.

[2] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Reports,* vol. 7, pp. 8176-8186, 2021.

[3] M. Fahad, H. Airf, A. Kumar, and H. K. Hussain, "Securing against apts: Advancements in detection and mitigation," *BIN: Bulletin Of Informatics,* vol. 1, no. 2, 2023.

[4] A. Wolsey, "The state-of-the-art in AI-based malware detection techniques: a review," *arXiv preprint arXiv:2210.11239,* 2022.

[5] L. Wang, C. Sun, C. Zhang, W. Nie, and K. Huang, "Application of knowledge graph in software engineering field: A systematic literature review," *Information and Software Technology,* vol. 164, p. 107327, 2023.

[6] I. P. Saputra, E. Utami, and A. H. Muhammad, "Comparison of anomaly based and signature based methods in detection of scanning vulnerability," in *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2022: IEEE, pp. 221-225.

[7] A. Aboalassaad and T. Malik, "The Impact of IPS and Firewall Placement on Network Security and Performance," in *2024 Cyber Research Conference-Ireland (Cyber-RCI)*, 2024: IEEE, pp. 1-7.

[8] R. Al-Amri, A. Hadi, A. H. Mousa, H. Hasan, and M. Kadhim, "The development of a deep learning model for predicting stock prices," *Journal of Advanced Research in Applied Sciences and Engineering Technology,* vol. 31, no. 3, pp. 208-219, 2023.

[9] V. Hassija *et al.*, "Interpreting black-box models: a review on explainable artificial intelligence," *Cognitive Computation,* vol. 16, no. 1, pp. 45-74, 2024.
[10] R. M. Al-Amri, A. A. Hadi, M. S. Kadhim, A. H. Mousa, A. Z. Matloob, and H. F. Hasan, "Enhancement of The Performance of Machine Learning Algorithms to Rival Deep Learning Algorithms in Predicting Stock Prices," *Babylonian Journal of Artificial Intelligence,* vol. 2024, pp. 118-127, 2024.
[11] H. Alqahtani and G. Kumar, "Advances in artificial intelligence for detecting algorithmically generated domains: Current trends and future prospects," *Engineering Applications of Artificial Intelligence,* vol. 138, p. 109410, 2024.
[12] P. Bhat, S. Behal, and K. Dutta, "A system call-based android malware detection approach with homogeneous & eterogeneous ensemble machine learning," *Computers & Security,* vol. 130, p. 103277, 2023.
[13] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "A survey on malware detection with graph representation learning," *ACM Computing Surveys,* vol. 56, no. 11, pp. 1-36, 2024.
[14] B. TRISTAN, N. EL MADHOUN, K. AL AGHA, and A. ZOUAOUI, "A Survey on Malware Detection with Graph Representation Learning," *arXiv preprint arXiv:2303.16004,* 2023.
[15] M. A. Talukder *et al.*, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *Journal of big data,* vol. 11, no. 1, p. 33, 2024.
[16] M. Schmitt, "Securing the digital world: Protecting smart infrastructures and digital industries with artificial intelligence (AI)-enabled malware and intrusion detection," Journal of Industrial Information Integration, vol. 36, p. 100520, 2023.
[17] R. M. Al-Amri, D. N. Hamood, and A. K. Farhan, "Theoretical background of cryptography," *Mesopotamian Journal of CyberSecurity,* vol. 2023, pp. 7-15, 2023.
[18] A. A. Hashim and A. H. Mousa, "An evaluation framework for diabetes prediction techniques using machine learning," in *BIO Web of Conferences*, 2024, vol. 97: EDP Sciences, p. 00125.
[19] M. E. Manaa, F. J. Abd Al-Razaq, and H. A. Al-Khamees, "Enhancing IoT Security: An Optimization Algorithm for Fog Layer-Based DDoS Attack Mitigation Framework," *Iraqi Journal of Science,* 2025.
[20] A. Gaurav, B. B. Gupta, and P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," *Enterprise Information Systems,* vol. 17, no. 3, p. 2023764, 2023.
[21] Z. Chen, E. Brophy, and T. Ward, "Malware classification using static disassembly and machine learning," *arXiv preprint arXiv:2201.07649,* 2021.
[22] S. M. Ali, R. M. Al-Amri, and A. K. Farhan, "Generation of Dynamic Substitution Boxes Using HSM Chaos System for Application in Color Images Encrypting," *International Journal of Intelligent Engineering & Systems,* vol. 16, no. 6, 2023.
[23] M. Ali, S. Shiaeles, M. Papadaki, and B. Ghita, "Agent-based Vs Agent-less Sandbox for Dynamic Behavioral Analysis," 2019.
[24] D. Trizna, L. Demetrio, B. Biggio, and F. Roli, "Nebula: Self-attention for dynamic malware analysis," *IEEE Transactions on Information Forensics and Security,* 2024.
[25] G. Karat, J. M. Kannimoola, N. Nair, A. Vazhayil, S. VG, and P. Poornachandran, "CNN-LSTM hybrid model for enhanced malware analysis and detection," *Procedia Computer Science,* vol. 233, pp. 492-503, 2024.
[26] H. Jmila and M. I. Khedher, "Adversarial machine learning for network intrusion detection: A comparative study," *Computer Networks,* vol. 214, p. 109073, 2022.
[27] C. P. Chenet, A. Savino, and S. Di Carlo, "A survey on hardware-based malware detection approaches," *IEEE Access,* 2024.
[28] Y. Li, W. Zhao, Y. Su, W. Li, and C. Yuan, "Overview of cloud computing deployment mode and technology development trend," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2023: IEEE, pp. 1-5.
[29] M. Ali, S. Shiaeles, M. Papadaki, and B. V. Ghita, "Agent-based vs agent-less sandbox for dynamic behavioral analysis," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 2018: IEEE, pp. 1-5.
[30] C. Li *et al.*, "DMalNet: Dynamic malware analysis based on API feature engineering and graph learning," *Computers & Security,* vol. 122, p. 102872, 2022.
[31] Z. Dai *et al.*, "An intrusion detection model to detect zero-day attacks in unseen data using machine learning," *PloS one,* vol. 19, no. 9, p. e0308469, 2024.
[32] S. Kumar, P. Ahlawat, and J. Sahni, "IOT malware detection using static and dynamic analysis techniques: A systematic literature review," *Security and Privacy,* vol. 7, no. 6, p. e444, 2024.
[33] A. Q. Raheema, M. Kamalrudin, and N. R. Dzakiyullah, "Multi-Level Fusion for Enhanced Host-based Malware Detection in ICT-Enabled Smart Cities," *Fusion: Practice & Applications,* vol. 15, no. 2, 2024.

[34] Y. Chang *et al.*, "RedStone: Curating general, code, math, and QA data for large language models," *arXiv preprint arXiv:2412.03398,* 2024.
[35] A. K. Ghazi-Tehrani and H. N. Pontell, "Phishing evolves: Analyzing the enduring cybercrime," in *The New Technology of Financial Crime*: Routledge, 2022, pp. 35-61.
[36] H. A. Noman, Q. Al-Maatouk, and S. A. Noman, "A static analysis tool for malware detection," in *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2021: IEEE, pp. 661-665.
[37] M. V. S. Babu and K. Banana, "A study on narrow artificial intelligence—An overview," *Int. J. Eng. Sci. Adv. Technol,* vol. 24, pp. 210-219, 2024.
[38] M. Ashik *et al.*, "Detection of malicious software by analyzing distinct artifacts using machine learning and deep learning algorithms," *Electronics,* vol. 10, no. 14, p. 1694, 2021.
[39] E. E. Abdallah and A. F. Otoom, "Intrusion detection systems using supervised machine learning techniques: a survey," *Procedia Computer Science,* vol. 201, pp. 205-212, 2022.
[40] A. Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," *Journal of Big Data,* vol. 8, no. 1, p. 90, 2021.
[41] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics,* vol. 61, p. 101182, 2021.
[42] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU," *arXiv preprint arXiv:2305.17473,* 2023.
[43] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," IEEE Access, vol. 9, pp. 157727-157760, 2021.
[44] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," *International journal of information security,* vol. 22, no. 5, pp. 1125-1162, 2023.
[45] H. Satilmiş, S. Akleylek, and Z. Y. Tok, "A systematic literature review on host-based intrusion detection systems," *Ieee Access,* vol. 12, pp. 27237-27266, 2024.
[46] H. Satilmiş, S. Akleylek, and Z. Y. Tok, "A systematic literature review on host-based intrusion detection systems," *Ieee Access,* vol. 12, pp. 27237-27266, 2024.
[47] B. Nawaal, U. Haider, I. U. Khan, and M. Fayaz, "Signature-based intrusion detection system for IoT," in *Cyber Security for Next-Generation Computing Technologies*: CRC Press, 2024, pp. 141-158.
[48] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly-and signature-based IDS for network security using hybrid inference systems," *Mathematical Problems in Engineering,* vol. 2021, no. 1, p. 6639714, 2021.
[49] M. A. Alsoufi *et al.*, "Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review," *Applied sciences,* vol. 11, no. 18, p. 8383, 2021.
[50] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet of things,* vol. 3, no. 1, p. 5, 2023.
[51] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE access,* vol. 9, pp. 22351-22370, 2021.
[52] G. Karat, J. M. Kannimoola, N. Nair, A. Vazhayil, S. VG, and P. Poornachandran, "CNN-LSTM hybrid model for enhanced malware analysis and detection," *Procedia Computer Science,* vol. 233, pp. 492-503, 2024.
[53] J. S. Bailey and M. R. Burch, *Research methods in applied behavior analysis*. Routledge, 2024.
[54] A. Ouaguid, M. Ouzzif, and N. Abghour, "Vulnerability Detection Approaches on Application Behaviors in Mobile Environment," *arXiv preprint arXiv:2307.16064,* 2023.
[55] F. A. Vadhil, M. L. Salihi, and M. F. Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES International Journal of Artificial Intelligence,* vol. 13, no. 1, pp. 711-721, 2024.
[56] G. Singh and N. Khare, "A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques," *International Journal of Computers and Applications,* vol. 44, no. 7, pp. 659-669, 2022.
[57] A. Khanan, Y. A. Mohamed, A. H. H. Mohamed, and M. Bashir, "From bytes to insights: a systematic literature review on unraveling IDS datasets for enhanced cybersecurity understanding," *IEEE Access,* vol. 12, pp. 59289-59317, 2024.
[58] M. S. Iqbal, W. Ahmad, R. Alizadehsani, S. Hussain, and R. Rehman, "Breast cancer dataset, classification and detection using deep learning," in *Healthcare*, 2022, vol. 10, no. 12: MDPI, p. 2395.
[59] E. Alomari, I. Katib, A. Albeshri, T. Yigitcanlar, and R. Mehmood, "Iktishaf+: a big data tool with automatic labeling for road traffic social sensing and event detection using distributed machine learning," *Sensors,* vol. 21, no. 9, p. 2993, 2021.
[60] M. Franceschini, "A neural-network based anomaly detection system and a safety protocol to protect vehicular network," *arXiv preprint arXiv:2411.07013,* 2024.

[61] H. Yukhymenko, R. Staab, M. Vero, and M. Vechev, "A synthetic dataset for personal attribute inference," *Advances in Neural Information Processing Systems,* vol. 37, pp. 120735-120779, 2024.
[62] Ammara, D. A., Ding, J., & Tutschku, K. (2024). "Synthetic Data Generation in Cybersecurity: A Comparative Analysis." *arXiv preprint arXiv:2410.16326*.
[63] Li, H., Su, J., & Wang, K. (2025). "Advancing CAN Network Security through RBM-Based Synthetic Attack Data Generation for Intrusion Detection Systems." *arXiv preprint arXiv:2503.21496*.
[65] R. Deshmukh, "Malware Classification using Machine Learning and Deep Learning," Medium, Jul. 2021.
[66] X. Zhao, K. W. Fok, and V. L. L. Thing, "Enhancing Network Intrusion Detection Performance using Generative Adversarial Networks," arXiv preprint arXiv:2404.07464, Apr. 2024.
[67] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: An Improved Siam-IDS for Handling Class Imbalance in Network-Based Intrusion Detection Systems," arXiv preprint arXiv:2009.10940, Sep. 2020.
[68] H. J. Ahn, S. Sharma, R. Matam, and J. Park, "A comprehensive survey on Android malware detection using machine learning," IEEE Access, vol. 11, pp. 39193–39226, 2023. doi: 10.1109/ACCESS.2023.3260207.
[69] A. Kumar, S. Sharma, and R. Singh, "Static analysis framework for permission-based dataset generation and android malware detection using machine learning," EURASIP Journal on Information Security, vol. 2024, no. 33, pp. 1–17, Oct. 2024
[70] M. Nasir, A. R. Javed, M. A. Tariq, M. Asim, and T. Baker, "Feature engineering and deep learning-based intrusion detection framework for securing edge IoT," Journal of Supercomputing, vol. 78, pp. 1–15, 2022.
[71] I. Kim and T. M. Chung, "Malicious-Traffic Classification Using Deep Learning with Packet Bytes and Arrival Time," in Future Data and Security Engineering, Lecture Notes in Computer Science, vol. 12466, Springer, Cham, 2020, pp. 267–280. doi: 10.1007/978-3-030-63924-2_20.
[72] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," arXiv preprint arXiv:1912.09363, Dec. 2019.
[73] Y. Ren, D. Zhao, D. Luo, H. Ma, and P. Duan, "Global-local temporal convolutional network for traffic flow prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 5, pp. 1935–1944, May 2020. doi: 10.1109/TITS.2019.2919620.
[74] Z. Cui, K. Henrickson, R. Ke, Z. Pu, and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting," arXiv preprint arXiv:1802.07007, Feb. 2018.
[75] L. Zhao et al., "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," arXiv preprint arXiv:1811.05320, Nov. 2018.
[76] R. Madan and P. S. Mangipudi, "Multi-Step Internet Traffic Forecasting Models with Variable Forecast Horizons for Proactive Network Management," Sensors, vol. 24, no. 6, p. 1871, Mar. 2024. doi: 10.3390/s24061871.
[77] T. Theodoropoulos et al., "Multi-step Short Term Traffic Flow Forecasting Using Temporal and Spatial Data," in Advances in Data Science and Adaptive Analysis, vol. 12, no. 1, pp. 1–20, 2020. doi: 10.1142/S2424922X22500103.
[78] O. A. Madamidola, F. Ngobigha, and A. Ez-zizi, "Detecting new obfuscated malware variants: A lightweight and interpretable machine learning approach," Intelligent Systems with Applications, vol. 25, Mar. 2025.
[79] M. A. Hossain and M. S. Islam, "Enhanced detection of obfuscated malware in memory dumps: a machine learning approach for advanced cybersecurity," Cybersecurity, vol. 7, Art. no. 16, Jan. 2024.
[80] X. Wei, Z. Cheng, N. Li, Q. Lv, Z. Yu, and D. Sun, "DWFS-Obfuscation: Dynamic Weighted Feature Selection for Robust Malware Familial Classification under Obfuscation," arXiv preprint arXiv:2504.07590, Apr. 2025.
[81] K. A. Dhanya, D. O. K., G. K. T., and P. Vinod, "Detection of Obfuscated Mobile Malware with Machine Learning and Deep Learning Models," in Machine Learning and Metaheuristics Algorithms, and Applications, Springer, Singapore, 2021, pp. 245–258.
[82] J. Zhu, J. Jang-Jaccard, A. Singh, P. A. Watters, and S. Camtepe, "Task-Aware Meta Learning-based Siamese Neural Network for Classifying Obfuscated Malware," arXiv preprint arXiv:2110.13409, Oct. 2021.
[83] E. Alhajjar, P. Maxwell, and N. D. Bastian, "Adversarial Machine Learning in Network Intrusion Detection Systems," arXiv preprint arXiv:2004.11898, 2020.
[84] B. Tafreshian and S. Zhang, "A Defensive Framework Against Adversarial Attacks on Machine Learning-Based Network Intrusion Detection Systems," arXiv preprint arXiv:2502.15561, 2025.
[85] X. Zhao, K. W. Fok, and V. L. L. Thing, "Enhancing Network Intrusion Detection Performance using Generative Adversarial Networks," arXiv preprint arXiv:2404.07464, 2024.
[86] H. A. Alatwi and C. Morisset, "Adversarial Machine Learning In Network Intrusion Detection Domain: A Systematic Review," arXiv preprint arXiv:2112.03315, 2021.

[87] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 3–18, doi: 10.1109/SP.2017.41.
[88] H. Hu et al., "Membership Inference Attacks on Machine Learning: A Survey," arXiv preprint arXiv:2103.07853, 2021.
[89] L. Song and P. Mittal, "Systematic Evaluation of Privacy Risks of Machine Learning Models," arXiv preprint arXiv:2003.10595, 2020.
[90] A. Veale, R. Binns, and L. Edwards, "Algorithms that remember: Model inversion attacks and data protection law," Philosophical Transactions of the Royal Society A, vol. 376, no. 2133, 2018, doi: 10.1098/rsta.2018.0083.
[91] I. Debicha et al., "Adv-Bot: Realistic Adversarial Botnet Attacks against Network Intrusion Detection Systems," Computers & Security, vol. 129, 2023, Art. no. 103176, doi: 10.1016/j.cose.2023.103176.
[92] H. A. Alatwi and C. Morisset, "Adversarial Machine Learning in Network Intrusion Detection Domain: A Systematic Review," arXiv preprint arXiv:2112.03315, 2021.
[93] K. Jarmul, "Privacy Attacks on Machine Learning Models," InfoQ, Aug. 6, 2019.
[94] M. Roesch, "Snort - Lightweight Intrusion Detection for Networks," in Proceedings of the 13th USENIX Conference on System Administration (LISA '99), Seattle, WA, USA, 1999, pp. 229–238.
[95] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report, Department of Computer Engineering, Chalmers University of Technology, 2000.
[96] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines," in Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN), Honolulu, HI, USA, 2002, vol. 2, pp. 1702–1707, doi: 10.1109/IJCNN.2002.1007774.