symmetry

MDPI

Article

Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation

Amir Djenna [1,*], Ahmed Bouridane [2], Saddaf Rubab [3] and Ibrahim Moussa Marou [1]

[1] College of New Technologies of Information and Communication, University of Constantine 2, Constantine 25000, Algeria
[2] Centre for Data Analytics and Cybersecurity, University of Sharjah, Sharjah 27272, United Arab Emirates
[3] Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, Sharjah 27272, United Arab Emirates
* Correspondence: amir.djenna@univ-constantine2.dz

Abstract: Malware, a lethal weapon of cyber attackers, is becoming increasingly sophisticated, with rapid deployment and self-propagation. In addition, modern malware is one of the most devastating forms of cybercrime, as it can avoid detection, make digital forensics investigation in near real-time impossible, and the impact of advanced evasion strategies can be severe and far-reaching. This makes it necessary to detect it in a timely and autonomous manner for effective analysis. This work proposes a new systematic approach to identifying modern malware using dynamic deep learning-based methods combined with heuristic approaches to classify and detect five modern malware families: adware, Radware, rootkit, SMS malware, and ransomware. Our symmetry investigation in artificial intelligence and cybersecurity analytics will enhance malware detection, analysis, and mitigation abilities to provide resilient cyber systems against cyber threats. We validated our approach using a dataset that specifically contains recent malicious software to demonstrate that the model achieves its goals and responds to real-world requirements in terms of effectiveness and efficiency. The experimental results indicate that the combination of behavior-based deep learning and heuristic-based approaches for malware detection and classification outperforms the use of static deep learning methods.

Keywords: cybersecurity analytics; digital forensics investigation; malware detection/mitigation; artificial intelligence

check for updates

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

Academic Editor: Alexander Shelupanov

Received: 1 February 2023
Revised: 15 February 2023
Accepted: 21 February 2023
Published: 8 March 2023

1. Introduction

The growing influence of telecommunication networks and the metaphor of the internet have revolutionized the way organizations carry out their activities. Indeed, the spectacular evolution of technology, digitalization [1], cloud/fog/edge computing [2–4], quantum computing [5], and the deployment of an exorbitant number of connected objects [6] have given rise to unprecedented cybercriminal activities. The growing threats of stealthy cyberattacks on critical infrastructures [7], data centers, government organizations, and financial sectors represent major challenges (from an individual and societal point of view). Complex cyberattacks rely on malicious software, also known as malware [8], intended for financial theft, cyber espionage, disruption, identity theft, exfiltration of sensitive data, and other political motivations [9].

Presently, the world is experiencing phenomenal advances in the computer field; however, the internationalization of cybercrime has also increased. In particular, during the COVID-19 pandemic, the heavy reliance on digital systems has led to a high increase of malware, such as ransomware [10]. COVID-19 has played an essential role in the proliferation of cyber threats [11], while organizations and individuals struggle to adapt to unpredictable new norms, further arming cyber attackers with modernized tools and techniques to execute more dangerous cyberattacks.

The current landscape of cyberattacks has led to an era of cyberwarfare [12], with no limits or boundaries, and with very active cyber espionage (sometimes between states or even between sovereign bodies). One of the most effective defenses used in the context of cyberwarfare mainly focuses on the use of modern malware.

According to the Kaspersky Security Bulletin (KSB) [13], in 2022, cybercriminals attacked users with 400.000 new malicious files daily (5% growth compared to 2021). Kaspersky's security researchers also discovered that the share of ransomware encountered daily increased by 181% compared to 2021, reaching 9500 encrypting files per day. Kaspersky security experts identified a 10% increase in the share of malicious files targeting the Android platform every day. The infamous 2022 campaigns, i.e., Harly [14] and the Triada Trojan [15], ambushed thousands of Android users around the world and are prime examples of this trend. The 2022 SpyCloud Ransomware Defense Report [16] surveyed over 300 individuals in active IT security roles at US, UK, and Canadian organizations, with at least 500 employees rating the threat of ransomware in 2022. The survey revealed that 90% of organizations were affected by ransomware in 2022 [17], which is a significant increase when compared to 2021, where the percentage was 72.5%.

Symantec [18] indicated that more than 50% of new malware are actually variants of existing ones.

In addition, the AV-TEST Institute registers more than 450,000 new malicious programs (malware) every day, as well as potentially unwanted applications (PUAs) [19]. Figure 1 shows the total amount of malware and PUAs for the year 2022:

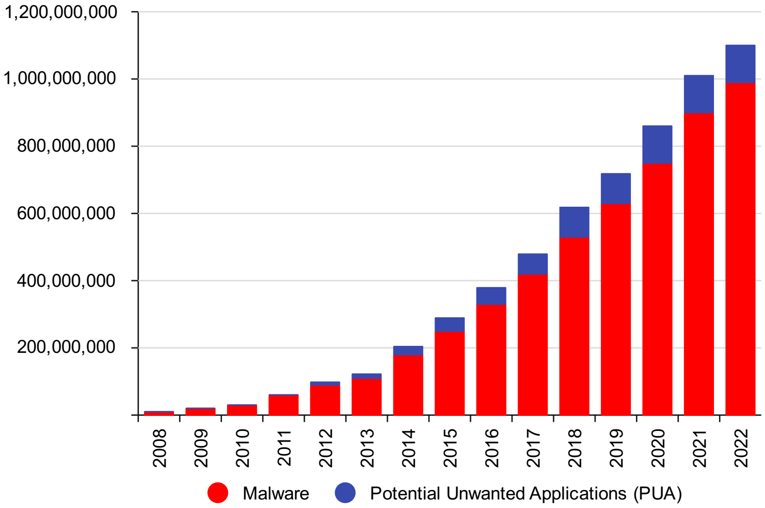


Figure 1. Total amount of malware and PUAs from 2008 to 2022.

Cyberattackers attempt to exploit known and unknown vulnerabilities to develop successful infections. With the massive deployment of an exorbitant number of connected objects, it is easy to imagine the scale of malware attacks that can be launched on these devices. This results in a major challenge that needs to be addressed.

In this increasingly digital world, the healthy functioning of ecosystems depends on the reliability and security of services, operations, and transactions that are ensured using encryption algorithms. However, this approach is inefficient when using a conventional computer. This implies that innovative solutions are need to improve classical cryptography. Quantum cryptography (which is a combination of quantum mechanics and classical cryptography with unconditional security) and quantum key distribution (QKD) could be designed to deal with quantum communication attacks. Indeed, quantum cryptography is based on the transmission of randomly generated qubits, which ensures the inviolability of exchanges under all circumstances [20,21]. These qubits constitute keys are used in encryption protocols. To send qubits over long distances, the preferred medium is the photon, which allows the encoding of information on observable variables, such as the polarization of light.

Moreover, quantum technology, such as quantum communications (ensuring the inviolability of information communicated along network fiber channels) and quantum computers (designed to perform tasks much more accurately and efficiently than conventional computers [22], which should lead to computers able to perform certain calculations efficiently) could provide new opportunities for the development of quantum machine learning and can help to achieve more secure communications [23]. Given the constraints of time, data volume, and complexity, quantum technology has the potential to enhance AI capabilities and serve as an accelerator for innovation by processing large sets of data, solving complex problems faster, and integrating multiple sets of data. Quantum technology could be a promising solution to bringing artificial intelligence into a new era in terms of execution speed and huge amounts of data processing, enabling AI to tackle more complex problems. Such a breakthrough could provide unprecedented momentum to many problems that require intensive calculations, which is becoming increasingly difficult as more complex data and relationships are added within the variables. For this purpose, large-scale quantum (LSQ) combined with artificial intelligence would be a major revolution for cybersecurity. Therefore, leveraging massive quantum computing capabilities could provide a strategic superpower for cybercriminals to perform adversarial cyber activities with devastating impacts.

Figure 2 shows the distribution of malware from January 2022 to June 2022. We observed 6% IoT malware, 19% crypto-jacking, 28% malicious intrusions, 53% malicious office and PDFs, 105% ransomware, and 167% encrypted threats [24].

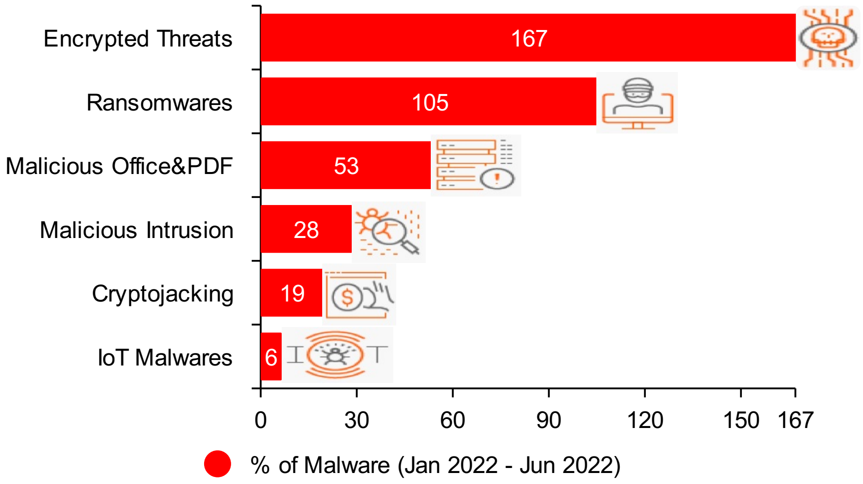


Figure 2. Global Malware trends by 2022.

It is imperative to develop resilient defense mechanisms against cyberattacks based on known and unknown malware. However, due to the multitude of samples and families of malware, it is hard to provide a rapid and automatic response in real time to deal with modern malware issues. Therefore, one must integrate advanced artificial intelligence (AI) techniques, such as deep learning (DL), which have been proven useful in other applications. Moreover, behavioral approaches can be studied through probabilistic reasoning, and diverse solutions can be provided for complex problems [25].

Given that cybersecurity experts are strongly considering artificial intelligence as a potential field and a prime attraction topic to improve conventional cybersecurity solutions, traditional machine learning techniques, such as RF (random forest), DT (decision tree), and SVM (support vector machine) have been widely used used in cyber security to classify malware. However, almost all methods have several shortcomings. Technical experience shows that these methods are limited and do not work efficiently for zero-day or unknown malware. Thereafter, the research paradigm shifted from traditional machine learning methods to deep learning-based methods to better identify malware. The results are much better improved compared to ML methods; moreover, it was observed that the average time required to detect malware decreases with the intervention of deep learning-based methods compared to traditional machine learning-based methods. The accuracy is improved as well even with large amounts of data. Nevertheless, deep learning methods come with certain limitations. Deep learning-based methods often face difficulties in identifying and analyzing different variants of malware in real time. These methods are also not suitable for dealing with malware evasion techniques and may not produce efficient zero-day detection systems. Therefore, there is a need for reliable and efficient solutions to enhance cybersecurity analytics, digital forensics investigation, and cyber defense capabilities by using dynamic deep learning-based methods. This is especially important since modern malware is continuously growing in sophistication and number, which could lead to chaos in critical infrastructure. The main motivation of this work is to develop the ability to detect, analyze, and identify both known and unknown malware in real time with several variants of modern malware. This will help advance cybersecurity measures.

Four pertinent questions should be asked:

1. What types of threats are produced by malware?
2. What types of vulnerabilities are exploited by malware?
3. Which cyberattack vectors are generated by malware?
4. Which malware and operating modes are used to violate the security properties?

Overall, in this work, we:

- Provide a classification of modern malware;
- Identify the top cybersecurity micro-domains that are necessary to maintain the cybersecurity posture of cyberspace;
- Perform an in-depth analysis of a specific dataset related to real malware traffic;
- Provide a taxonomy of potential DL techniques that may be used to enhance cybersecurity solution;
- Develop a new cybersecurity approach based on a dynamic deep learning model for the autonomous detection and mitigation of modern (known and unknown) malware.

The paper is organized as follows: Section 2 describes the importance of moving from classical malware detection and analysis to smart and autonomous detection/analysis through the incorporation of advanced AI techniques. Moreover, we provide a classification of modern malware based on famous samples and high-profile cases. Section 3 highlights the most common related work for malware detection and analysis. Section 4 presents the proposed work and our methodology to deal with modern malware detection and mitigation. The experimental results are described in Section 5 followed by some discussions. Finally, Section 6 presents our conclusion and some future works.

2. Background

To offer resilient and effective cybersecurity solutions to deal with modern malware detection and analysis with faster response and real-time automation, it is important to understand the fundamental approaches employed as well as the malware obfuscation techniques adopted by cyberattackers. The micro-domains of cybersecurity mainly revolve around intrusion detection/prevention, digital forensics, cyber threat intelligence, cybersecurity analytics, and malware detection/analysis. Figure 3 presents the top cybersecurity micro-domains that have emerged, where various AI techniques can be used to flesh out issues that arise in these fields of cybersecurity.

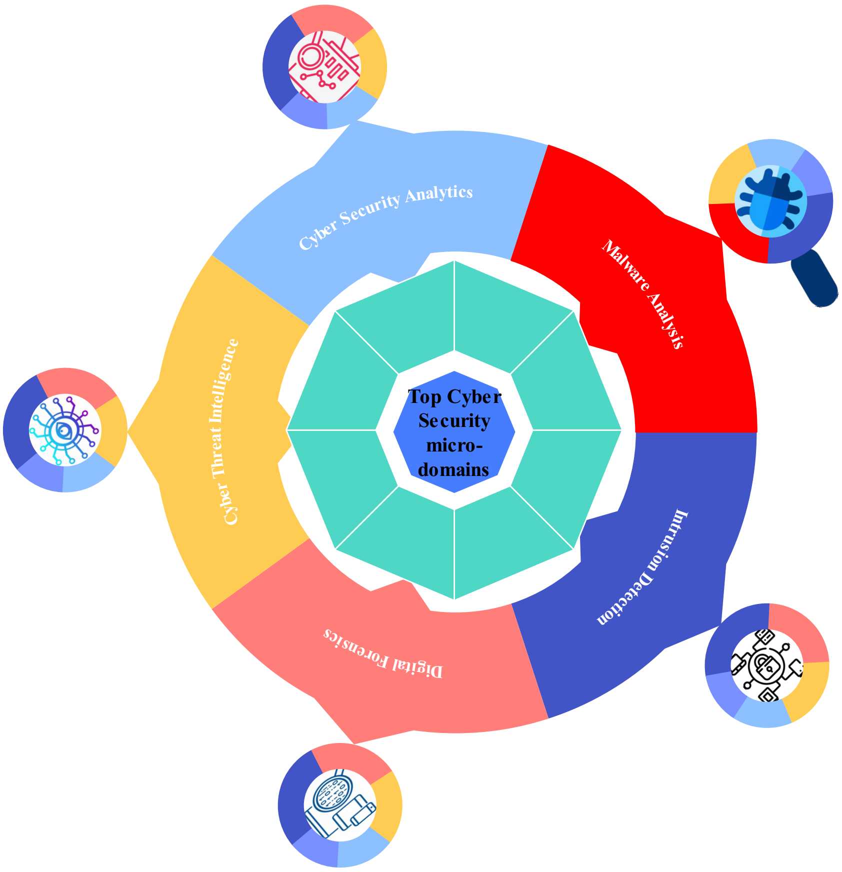


Figure 3. Top cybersecurity micro-domains/applications.

Given that cyberattack surfaces are expanding and cyber threats are becoming more complex, a new vision of cybersecurity must be established. For this purpose:

Cyber security analytics (CSA) is a proactive approach to cybersecurity that uses data collection, aggregation, correlation, and analysis capabilities to perform critical security functions that detect, analyze, and neutralize cyber threats and vulnerabilities before an attack occurs. Furthermore, it could be used for the detection, prevention, and mitigation of social engineering, APT (advanced persistent threats), modern and advanced malware, DDoS cyberattacks, unpatched vulnerabilities, and weak credentials.

Cyber threat intelligence (CTI) is a discipline based on intelligence techniques; it aims to collect and organize all information related to cyber threats in cyberspace in order to draw a cartography of cyberattackers and highlight trends. In other words, CTI refers to the process of information gathering on a potential threat, processing and analyzing data to better understand threats. Cyber threat intelligence is often split into three categories: strategic threat intelligence, tactical threat intelligence, and operational threat intelligence. Moreover, cyber threat intelligence operates on a life cycle, which involves six stages: direction, collection, processing, analysis, dissemination, and feedback.

Digital forensics (DF) is a branch of forensic science that focuses on the recovery and investigation of material found in digital devices related to cybercrimes. The process focuses on techniques for collecting and using traces (initially electronic) that can be recorded on very different types of media. The methodology uses nine phases for digital forensics to be acceptable to track cybercriminals: first response, search and seizure, collect the evidence, secure the evidence, data acquisition, data analysis, evidence assessment, documentation and reporting, and testifying as an expert witness.

Intrusion detection/prevention is a system that continuously monitors the network to identify potential incidents. The system records related information in logs, resolves incidents, and reports them to security administrators. Typically, intrusion detection/prevention should send alarms to administrators, drop the malicious packets, block bad traffic from the source address, reset the connection, and self-configure to prevent future intrusions. There are several types of intrusion detection/prevention that can be deployed for different purposes, such as network intrusion prevention, host intrusion prevention, network behavior analysis, and wireless intrusion prevention.

Malware detection/analysis is the process of identifying malicious software and unwanted object functioning and their impacts. This makes it possible to recover indicators of compromise to detect infected machines and, hence, anticipate future infections, study their impacts, identify exploited vulnerabilities, and identify the origin. The practice consists of determining and analyzing in-depth suspicious files, codes, and records on endpoints.

2.1. Malware Classification

Malware (or malicious software), as one of the major cybersecurity threats today, is a program or part of a program intended to disrupt, alter, or destroy all or part of the software elements that are essential to the proper functioning of a computer system, device, service, or network. Malware has been threatening us more in the past few years, with millions of malware samples observed in recent years.

Malware was born in the 1970s (it was named Creeper); it could connect to a remote system using a modem and display the following error message: “I'M THE CREEPER: CATCH ME IF YOU CAN”. Malware has evolved to the point where it is now able to modify the rotation speed of a nuclear centrifuge (e.g., such as what Stuxnet [26] malware did to an Iranian nuclear power plant in 2010); it can steal sensitive information (as was the case with Flamer [27] in 2012); or use satellite links to communicate with the attacker (as was the case with Turla [28] in 2015). WannaCry [29], discovered in May 2017, is one of the largest ransomware attacks in history, having infected over 230,000 Windows PCs in 150 countries, many of which belong to government agencies and hospitals. WannaCry spread using a Windows vulnerability named MS17-010 [30]. Although the attack was halted in May 2017, WannaCry has not been completely eradicated. In March 2018, Boeing was targeted, and further cyberattacks remain possible. In addition, other ransomware strains that exploit the same Windows vulnerability have been developed, such as Petya [31] (Petya.A, Petya.B, or PetrWrap) and NotPetya [32]. On 3 December 2018, Samsam [33], also known as MSIL/Samas.A, targeted industries, some of which were critical infrastructure. Cyberattackers used the JexBoss exploit kit to gain access to vulnerable JBoss applications, remote desktop protocol (RDP) to gain persistent access to victim networks, and brute force attacks. Thereafter, the authors of Samsam escalated privileges for administrator rights, dropped malware on the server and ran a corrupted executable file, all without the victim's permission. This gave them the ability to perform malicious actions, such as opening an email or visiting a compromised website or redirecting and infecting via RDP with minimal detection. Recently, on 5 July 2021, Darkside [34], a ransomware-as-a-service (RaaS), ran a dynamic link library (DLL) program used to delete volume shadow copies available on the system. The malware collected, encrypted, and sent system information to the threat actor's command and control (C&C) centers, and generated a ransom note for the victim.

Therefore, malware has different functionalities, and it can be classified by family and sample. However, it is important to classify malware according to its impact and goals due to the diversification of malware samples. In this context, Figure 4 illustrates a classification of modern malware types with examples that have occurred.

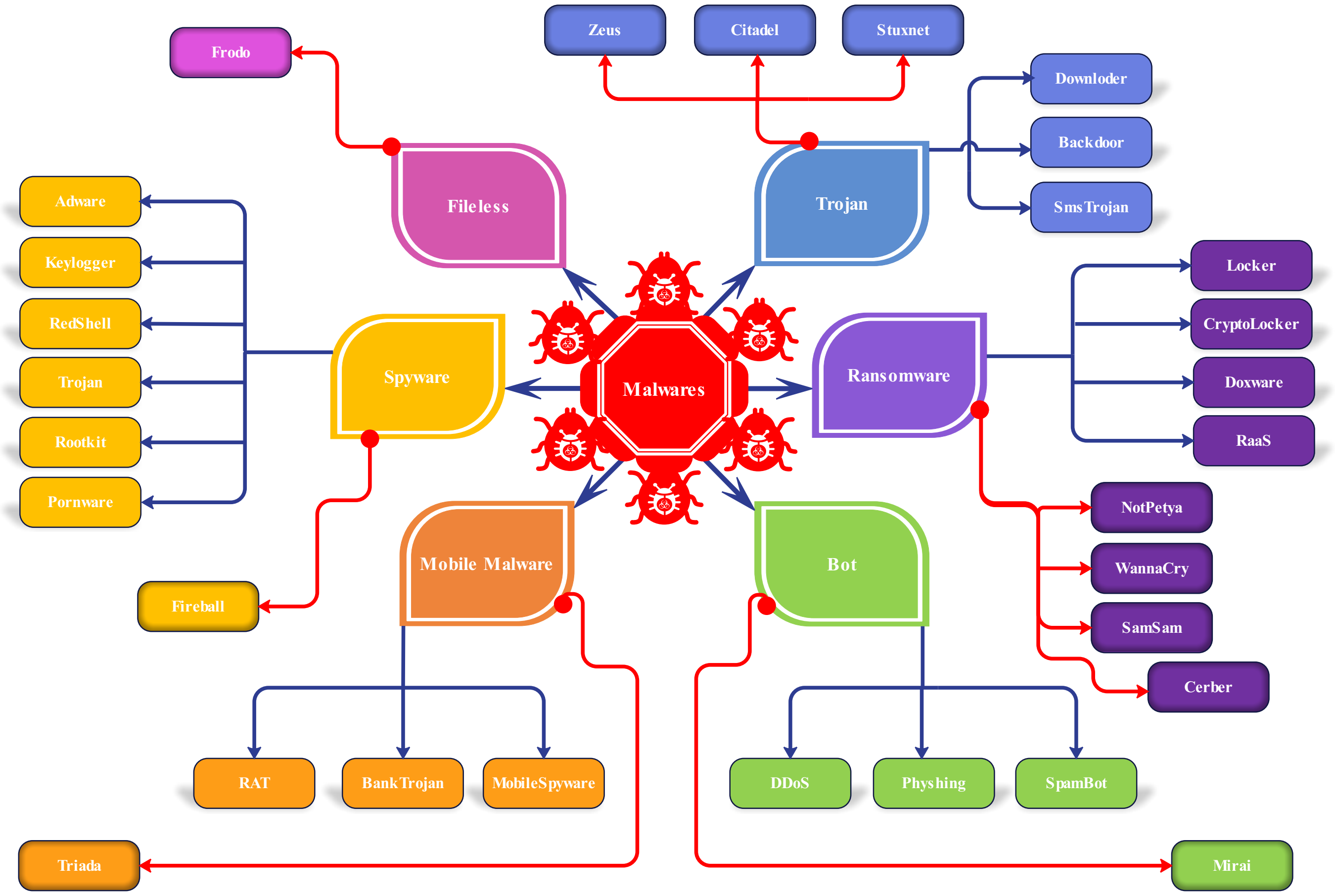


Figure 4. Modern malware taxonomy.

A virus is a type of malware. It belongs to a viral attack type and is a malicious program designed to reproduce itself by infecting other programs (and, thus, spreading from device to device). Once infected, the virus will seek to damage, alter, or delete files or data on electronic devices. It is transmittable, and it can have delayed actions. It spreads mainly through the internet, USB drives, and email. There are four main families of viruses: macro viruses, email viruses, polymorphic viruses, and Trojan viruses. Macro viruses are mostly found in documents or embedded as malicious code in word-processing software. They exploit the vulnerabilities of software applications. Email viruses use the macros of an attached document (in word or PDF). When opening the document, the macros are activated. Locally, they cause damage to the user's system. They are located in RAM and contaminate the files as they are executed. For example, they can take the form of a Windows driver file (.vxd). They are then loaded as soon as the system starts (before the antivirus is loaded). A polymorphic virus is designed to create copies of itself, each time changing the layouts and values of the bytes that make it up (in order to avoid antivirus detection). A Trojan horse is a type of software that presents itself in an honest, useful, or pleasant light; once installed on electronic devices, it performs hidden and harmful actions. Botnet is short for robot network. The term 'robot' or 'bot' is a generic term for an automated program that performs tasks without user intervention. Botnets are the largest threats to the internet today. A botnet is a network of compromised machines that can be remotely commanded, controlled, and coordinated by an attacker to fulfill a malicious directive [35]. A computer worm is a malicious program that originates on a single device and searches for other connected devices via a local network or internet connection. A keylogger is a software program that, when installed on the user's device, makes it possible to record the keystrokes entered by the user. A keylogger can be installed by the administrator of a company to control the activities of employees; it can also be installed by a hacker to obtain information about his victims. Spyware is a type of virus that installs itself in a system for the purpose of collecting and transferring a victim's sensitive information. Finally, ransomware is extortion software designed to block electronic devices (or an entire system), encrypt files, and then demand a ransom to recover the data. Ransomware can involve several methods for infecting systems, such as external remote services, zero-day exploits, phishing, botnets, and emails.

2.2. Traditional Malware Detection and Analysis

Malware analysis is a process of examining executable files, with the aim of extracting as much precious information as possible. The purpose of the data is to draw the perimeter of a cyberattack and to detect the different functionalities and behaviors of the malicious program. The purpose lies in preventing similar cyberattacks on information systems. There are three main axes of malware detection and analysis, as shown in Figure 5:

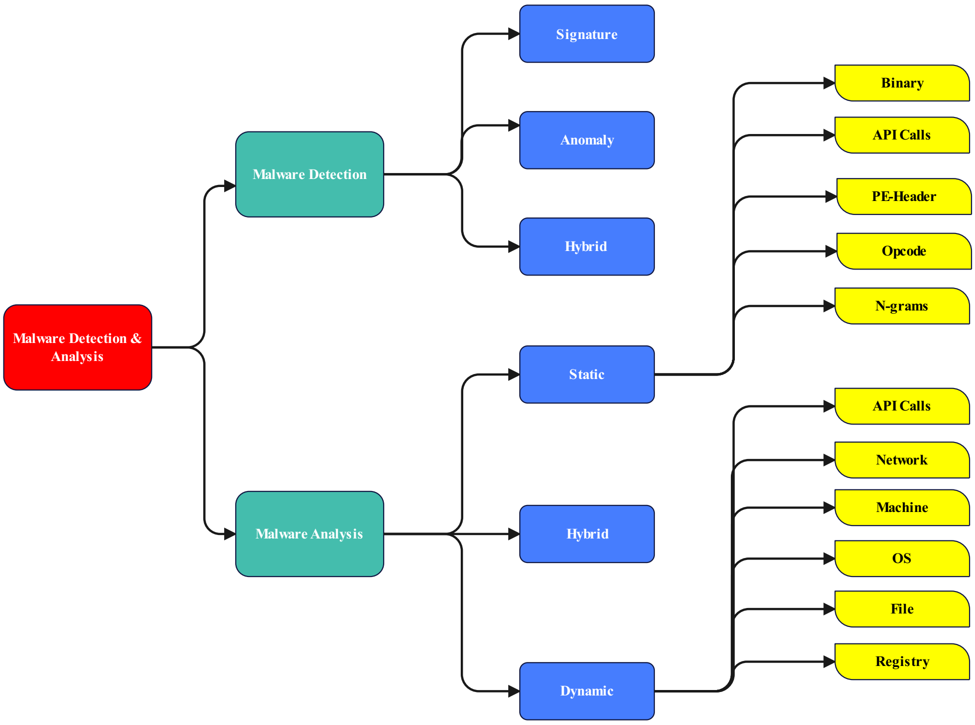


Figure 5. Malware detection and analysis.

2.2.1. Static Analysis

Static analysis is a method of analyzing malware without launching it. The goal is to extract most of the metadata. Although static analysis can easily analyze and detect known malware, it fails for complex and new malware. Indeed, malware developers use obfuscation techniques to hide the true nature of their applications or use polymorphism and metamorphism techniques to change the appearance of the code of different samples of malware. Some of the advanced static analysis methods can analyze complex malware, but these processes are quite cumbersome and require a lot of advanced knowledge of operating systems and disassembly. For instance, PE Explorer is a tool that allows examining Windows .exe and .dll files. Androguard is a popular static analysis tool for android applications. It has the ability to assess the similarity between two applications. Androguard compares the codes of the two applications and calculates which methods are identical, similar, and present in one but not in the other.

2.2.2. Dynamic Analysis

Dynamic analysis involves analyzing the behavior and effects of malware when it is executed. We have to understand what the malware does when it is executed. The goal is to collect real-time data related to malware behavior and its impact on the system. With this technique, we observe all the functionality of the malware and its effect on the environment during its execution. Usually, the file is executed in a virtual environment. Dynamic analysis is always preferred over static analysis because even if the structure of malware changes, the behavior and characteristics will never change and always remain the same, which helps dynamic analysis to detect malware easily. Wireshark [36] and TCP dump [37] are excellent tools that capture packets traveling through a network and analyze them. DroidBox is a sandbox tool for Android applications. During the execution of the application, the tool records the network communications carried out by the application, the accesses to files, the services launched, the classes loaded, the cryptographic operations via the Android API, the messages, and the outgoing calls.

2.2.3. Hybrid Analysis

A hybrid analysis is a technique that is composed of two malware analysis techniques, namely static and dynamic analysis.

2.2.4. Malware Detection Based on Signature

A signature is a piece of sequence injected into the application program by malware authors, which uniquely identifies malicious software. This is a fast and effective approach for known malware. However, the downside of this technique is its inefficiency against unknown attacks and, hence, it cannot detect new unknown malicious software because no signature is available for such attacks; worse still, malware developers can constantly modify their codes or the way in which they are wrapped, so that they do not produce identical signature to previous versions.

2.2.5. Malware Detection Based on Behavior

In this technique, program behavior is used to determine whether it is malicious or benign. A behavior-based detector goes through three phases, i.e., (1) information gathering, which involves collecting information about the malware. (2) Interpretation of the collected information to extract the most relevant details, grouping them together to create a signature (behavioral model). (3) The detection phase, which consists of finding a correspondence between the signature of the malware in question and that representing malicious behavior.

Most pioneering anti-virus vendors, such as Kaspersky, McAfee, Symantec, Avast, and others have developed anti-virus solutions to protect devices and legitimate users from malware. However, these solutions are designed by virtue of signature-based methods. Furthermore, modern malware uses evasive techniques to escape checkpoints, such as encryption, oligomorphism, polymorphism, metamorphism, stealth, and packaging, of which, malware authors design and hide zero-day malicious code. Hence, there is a need to design automated systems that are capable of detecting and classifying malware autonomously in a real-time manner, where AI could be the best for this challenge and for developing a next-generation antivirus.

In recent years, machine learning (ML) and deep learning (DL) have considerably brought dazzling advances in several fields of research. More specifically, artificial intelligence has experienced an extraordinary boom thanks to ML and DL.

ML is an attractive area of computer science that had been used successfully in searching, image recognition, and decision-making [38]. Moreover, DL is based on powerful and widely applicable models, allowing for the extraction of relevant information for complex tasks. For this purpose, DL could be a relevant contribution to malware detection, classification, analysis; Botnet identification, detection; cyber attacks mitigation; intrusion detection, prevention; incident response; network traffic analysis; APT detection; cybercriminals identification; deep packet inspection, and cybersecurity analytics. Figure 6 depicts a taxonomy on the potential application of ML models in several cybersecurity fields.

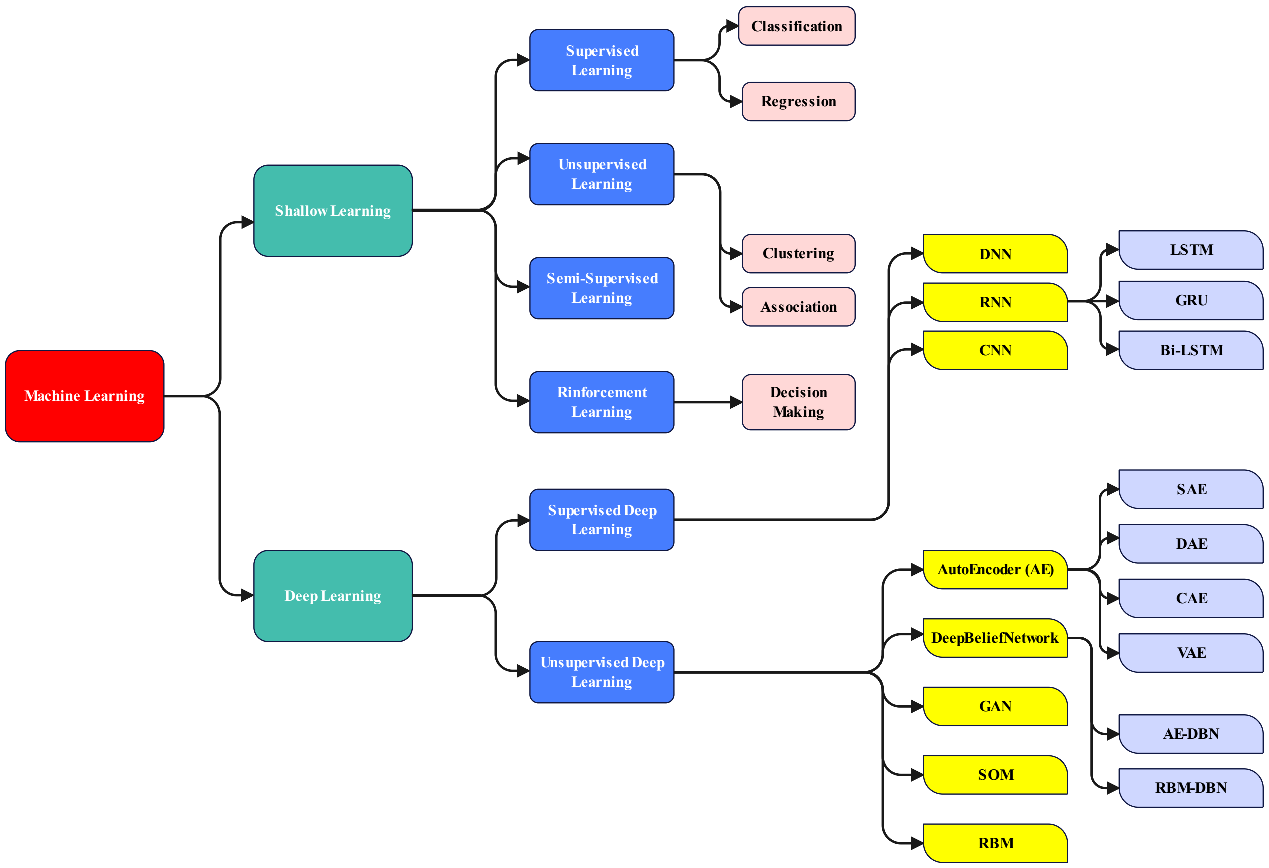


Figure 6. Machine learning-based approaches for cybersecurity.

3. Related Works

Cyberspace is vulnerable to a wide range of new-generation malware samples that can severely impact human life, safety, and business continuity. Several recent research works have been conducted to detect malware on Windows, Android, iOS, IoT, and APT applications. For instance, Jerlin et al. [39] implemented the first malware detection system based on machine learning techniques (naïve Bayesian algorithm). The authors studied different types of information contained in a PE file, such as character strings, APIs, and byte sequences.

The authors of [40] analyzed malware samples according to the source-and-sink method pairs extracted by FlowDroid. Catak et al. [41] proposed a static LSTM-based approach to detect five malware families, i.e., Trojan, rootkit, backdoor, virus, and worm. The authors of [42] focused on developing a system based on DL that uses CPU, memory, and battery usage to predict malware. Pajouh et al. [43] proposed a deep recurrent neural network-based deep learning system for android malware detection.

Karbab et al. [44] proposed an android malware detection system that uses the bag-of-words (BOW) model to extract various features of the application, such as hardware components, application components, network addresses, restricted API calls, suspicious API calls, requested permissions, used permissions, and filtered intents.

Huang et al. [45] developed a color-based CNN detection system for the Android platform. The system converted the Android classes.dex file to an RGB (red, green, blue) image. Zhu et al. [46] provided an Android malware framework called MSerNetDroid, which is based on a multi-head squeeze-and-excitation residual network to extract features from the manifest file permissions, API calls, and hardware features.

In 2022, Seraj et al. [47] put forward a fake android anti-malware-based MLP, static permissions, and applications to address android malware detection. Sasidharan et al. [48] proposed another contribution to android malware detection architecture that uses the profile hidden Markov model, encoded patterns, and log-likelihood score. Lee et al. [49] utilized a Chameleon–Hunter algorithm based on iOS Chameleon apps, UI-based illicit activity threats, and suspicious PHI-UI to solve some existing threats related to iOS mobile malware.

For windows malware detection, many researchers provided algorithms of classification-based ML features, such as AATR and ETW [50], Footprints [51], and Cuckoo sandbox based on SVM, Naïve Bayes, and RF [52].

For the detection of IoT malware using ML classifiers, recent research studies have contributed through ML-based IDS using botnet life cycle stage, wrapper methods, and channel-based features [53], an SDN-based framework using DDoS attack, SDNWISE customized controllers, IP packet counter, IP payload, and Cooja Simulator [54].

For APT malware detection using ML, some research works have proposed classification via XGB using ANOVA and variance feature selection [55], Bayesian Stackelberg Game using IoV, optimal mixed strategy, and DOBSS [56].

For ransomware detection using ML, several research works have introduced methods and models such as the ontology-driven framework (behavior-based using a knowledge-based system), API functions and behaviors [57], MFMCNS using majority voting rule for WannaCry and NotPetya [58], AMOEBA with memory-based using device-level storage and hardware accelerators [59], file-sharing traffic analysis using crypto-ransomware, ML and file activities [60], and DNAAact-Ran using digital DNA sequences [61]. Qiu et al. [62] developed a new framework based on multi-view feature intelligence to learn the presentation of a target capability from known malware families for recognizing unknown and evolving malware with the same capabilities. Qiao et al. [63] proposed a new method to detect adversarial ELF malware using model interpretation techniques. Xu et al. [64] proposed a novel slow-aging solution named SDAC to address the model aging problem in android malware detection using semantic distance-based API clustering. Fan et al. [65] developed FalDroid, a novel approach that automatically classifies Android malware and selects representative malware samples in accordance with 'fregraphs'.

Signature-based malware detection methods have been widely used. These methods work quickly and effectively against well-known malware. However, when it comes to unknown malware (zero-day), the results are not satisfactory; therefore, signature-based methods are not suitable for unknown malware detection. Thus, the lack of performance detection implies that signature-based methods are not suitable for modern malware detection. Moreover, new-generation malware is coded in a complex style and can evade security solutions, such as firewalls and anti-viruses that cannot deal with advanced evasion techniques. Moreover, the unknown signatures of modern malware require that security mechanisms should be proactive rather than reactive. Hence, the investigation into deep learning methods remains essential.

4. Proposed Work

Every year, many companies and organizations fall victim to malware attacks. A cyberattack-based malware can lead to devastating consequences, including financial loss, theft of sensitive information, compromised supply chains, and more. Antivirus scanners cannot meet protection needs, resulting in millions of hosts being attacked. Malware detection systems have been extensively researched and play an important role in cybersecurity. The objective of our study is to propose a systematic approach to classify modern malware on a new comprehensive dataset named CICAndMal2017, provided by the Canadian Institute for Cybersecurity (CIC) [66], using machine and deep learning algorithms. The performances of the proposed detection approaches were evaluated by taking into account the different evaluation measures.

4.1. Dataset

CICAndMal2017 is an advanced malware dataset that was used in our study. CICAndMal2017 [67] is provided by the Canadian Institute for Cybersecurity to train robust malware detection classifiers. It contains over 10.854 samples (4.354 malware and 6.500 benign) from several sources. The dataset contains four categories of modern malware (adware, SMS Malware, scareware, and ransomware) and a benign file, which in turn consists of three folders (benign2015, benign2016, and benign2017). Each type of malware (category) is composed of a set of malware families, each containing a set of data instances. The set of classes also contains 85 characteristics (features) which are labeled and characterized on the feed network traffic. Table 1 shows the details of the used dataset.

Table 1. Dataset content description.

Folders	Number of Folders	Number of Samples
Adware	10	104
Ransomware	10	101
Scareware	11	112
Rootkit	10	105
SMS Malware	11	109
Benign	3	1702

The dataset is not structured, so we have 6 files (benign and malware) composed of several samples: 1702 for the benign folder and 426 for the malware folders, in addition, the contents of the target variable are not the same in the different classes.

4.2. Data Analysis and Exploration

Data analysis and exploration are very important steps for our study, consisting of reformulating the dataset and understanding the different variables to define a modeling strategy.

4.3. Data Preparation

As mentioned previously, the dataset was not structured. In other words, whenever data were collected from different sources, they were collected in raw formats, which were not suitable for analysis. Hereinafter, we explain the process of structuring the dataset. We carried out our study on two classification methods (binary classification and multi-class classification). For the binary classification as shown in Table 2, we changed all of the labels of the different classes of malware folders into a single malware class and a single class for benign traffic.

Table 2. Data preparation for binary classification.

Classes	Initial	Labialization
Adware	Dwogin, Ewin, Feiwo, Gooligan, Kemoge, Koodous, Mobidash, Selfmite, Shuanet, Youmi.	Malware
Ransomware	Charger, Jisut, Koler, Lockerpin, Pletor, PornDroid, RansomeBo, Simplocker, SVpeng, Wannalocker.	Malware
Scareware	AdroidDefender, AdroidSpy, AVforAndroid, Avpass, FakeApp, FakeAppAl, FakeAV, FakejobOffer, FakeTaoBao, Penetho, VirusShield.	Malware
SMS Malware	Beanbot, Biige, Fakeinst, FakeMark, FakeNotify, Jifake, Mazarbot, Nandrobox, Plankton, SMSsniffer, Zson.	Malware
Benign	Benign2015, Benign2016, Bening 2017.	Benign

For multi-class classification, it was the same process; we randomly chose most of the files of our dataset. The benign class label was unchangeable; we formed a label for the adware class, a label for the ransomware class, a label for the scareware class, a label for the SMS malware class, and a label for the rootkit class.

4.4. Data Preprocessing

AI algorithms learn from the data given to them; therefore, if the data are of poor quality, wrong, or incomplete, the resulting algorithm will lead to bad classification performances since it is supposed to reflect what it sees within the data. For this reason, it is imperative that the data will be well-prepared before passing them through the machine. The data should be cleaned, filtered, and normalized; this is called data preprocessing. Our approach is based on deep learning techniques, in which the preprocessing step relies on feature transformation, normalization, extraction, and encoding.

4.5. Feature Extraction

Regarding binary and multi-class classification, we used embedded methods to train our raw data with the random forest algorithm to determine the relevance of our attributes. This algorithm was chosen for its performance and accuracy.

As a result, we can conclude that variables such as Bwd URG Flags, Bwd PSH Flags, Fwd URG Flags, RST Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg packets/Bulk, ECE Flag Count, Bwd Avg Bytes/Bulk, Fwd Avg packets/Bulk, Bwd Avg Bulk Rate, and Fwd Avg Bulk Rate are not relevant to our study dataset. Table 3 presents the list of selected features for our study.

Table 3. Selected attributes for models.

Features	Type	Features	Type
Source Port	Int	Bwd Packets/s	Float
Destination Port	Float	Min Packet Length	Float
Protocol	Float	Max Packet Length	Float
Total Length of Fwd Packets	Float	Packet Length Mean	Float
Total Length of Bwd Packets	Float	Packet Length Std	Float
Fwd Packet Length Max	Float	Packet Length Variance	Float
Fwd Packet Length Min	Float	FIN Flag Count	Float
Fwd Packet Length Mean	Float	SYN Flag Count	Float
Fwd Packet Length Std	Float	PSH Flag Count	Float
Bwd Packet Length Max	Float	ACK Flag Count	Float
Bwd Packet Length Min	Float	URG Flag Count	Float
Bwd Packet Length Mean	Float	Down/Up Ratio	Float
Bwd Packet Length Std	Float	Average Packet Size	Float
Flow Bytes/s	Float	Avg Fwd Segment Size	Float
Flow Packets/s	Float	Avg Bwd Segment Size	Float
Flow IAT Mean	Float	Fwd Header Length.1	Float
Flow IAT Std	Float	Subflow Fwd Packets	Float
Flow IAT Max	Float	Subflow Fwd Bytes	Float
Flow IAT Min	Float	Subflow Bwd Packets	Float
Fwd IAT Total	Float	Subflow Bwd Bytes	Float
Fwd IAT Mean	Float	Init_Win_bytes_forward	Float
Fwd IAT Std	Float	Init_Win_bytes_backward	Float
Fwd IAT Max	Float	act_data_pkt_fwd	Float
Fwd IAT Min	Float	min_seg_size_forward	Float
Bwd IAT Total	Float	Active Mean	Float
Bwd IAT Mean	Float	Active Std	Float
Bwd IAT Std	Float	Active Max	Float
Bwd IAT Max	Float	Active Min	Float
Bwd IAT Min	Float	Idle Mean	Float
Fwd PSH Flags	Float	Idle Std	Float
Fwd Header Length	Int	Idle Max	Float
Bwd Header Length	Int	Idle Min	Float
Fwd Packets/s	Float	Flow Duration	Int
Total Fwd Packets	Int	Total Backward Packets	Int

4.6. Imputation and Encoding

In the imputation process, we analyzed all data with NAN (not a number) or INF (infinite values). We also removed columns that did not have numerical values because malware can be created at any time by any machine against any victim machine. These columns are as follows: Source IP, Destination IP, Source Port, Destination Port, Timestamp, Date, IP, Time, and Addresses. Additionally, we removed variables that have no relevance to learning.

In the encoding step, we labeled our data as follows: for binary classification, 1 represents malware and 0 represents benign. For multi-class classification, 0 represents benign, 1 represents adware, 2 represents ransomware, 3 represents scareware, 4 represents SMS malware, and 5 represents rootkit.

4.7. Normalization

In this step, we placed all of the variables on the same scale, thus facilitating the learning of the models. We also divided all of our data into training data (Trainset) and testing data (Testset). As mentioned below, Table 4 and 5 gives details of the number of trains and tests for each classification:

Table 4. Data repartition of the dataset (binary classification).

Classification Type	Classes	Trainset	Testset
Binary	Benign	55,671	16,625
	Malware	886,461	302,638

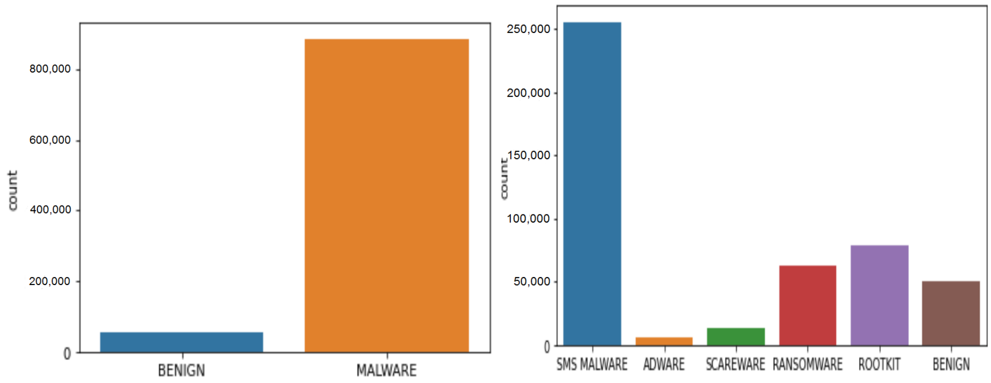
Table 5. Data repartition of the dataset (multi-class classification).

Classification Type	Classes	Trainset	Testset
Multiple	Benign	49,767	14,880
	Adware	5770	1341
	Ransomware	62,471	28,220
	Scareware	13,646	3930
	SMS Malware	255,487	72,816
	Rootkit	79,115	21,715

4.8. Data Visualization

The visualization of our target variables is given in Table 6:

Table 6. Label visualization for two classifications.

Binary and Multiple


As shown in Table 6, the dataset is unbalanced, and if we train the model without fixing this issue, the model will be completely biased and it impacts the feature correlation. To fix this, we used the sampling through the resampling technique, called oversampling, which is a process of generating synthetic data that attempts to randomly generate a sample of the attributes from observation in the minority class. We used the SMOTE (synthetic minority over-sampling technique) Python library to deal with the data imbalance in classification.

4.9. Model Architecture

We implemented two types of approaches for deep learning algorithms (CNN and DNN), and two types of approaches for machine learning algorithms (random forest and tree decision classifier) for both binary classification and multi-class classification. For all techniques, several tests were made to determine the best parameters. All methods were evaluated using a confusion matrix and diagrams that were necessary for the evaluation.

4.10. Malware Classification Based on Deep Learning Models

The CNN and DNN models that we implemented have some features in common, i.e., both models have an input layer, intermediate layers, and an output layer.

Each input layer of the two models takes the input of the size of the variables selected in the feature selection step. The intermediate layers have an activation function, and in our study, we used the ReLU function. The output layers have the same dimensions as the class numbers, which are determined based on the type of classification. For binary classification, we used the Sigmoid function as an activation function and the Softmax function for multi-class classification. In our study, the Sigmoid function generates numbers between -1 and 1. During development, we created a procedure to classify numbers greater than 0.5 as malware traffic and numbers less than 0.5 as benign traffic. The Softmax function gives a probability (the sum of which is equal to 1) at the output of each neuron, and the output neuron with the greatest probability enables us to decide that its associated class is the predicted class.

The models were compiled with a loss function, optimizer, and evaluation metric. For binary classification (malware/benign), we used binary cross-entropy as the loss function and categorical cross-entropy for multi-class classification. Both models contain the Adam optimizer with a learning rate of 0.01. We used the dropout technique to solve overfitting problems. Finally, we used early stopping to halt the training process when the validation loss is no longer improving.

4.11. Malware Classification Based on CNN

Usually, CNN is used for image detection. We used a 1D CNN model for malware classification. The 1D CNN works in the same way as 2D CNN or 3D CNN. We chose this model because of its high performance and learning rate. During our development of the 1D CNN model, we made some parameter adjustments to obtain better results. Figure 7 shows the architecture of our model on the 1D CNN for the two classifications. We transformed the dataset (training and evaluation data) into NumPy vectors of three variables ($\times 1$, $\times 2$, $\times 3$) with $\times 1$ being the number of observations, $\times 2$ being the number of variables in our study, and $\times 3$ initialized to 1. The model has the following layers: one-dimensional convolution layers whose first layers of the model represent the characteristics of the input data; and connecting layers, whose last layers represent the output.

4.12. Malware Classification Based on DNN

In recent years, DNNs (deep neural networks) have been considered some of the most important computational networks. The integration of DNNs in cybersecurity solutions, in general, and for the detection and analysis of malware, in particular, is intended as a relevant and promising research axis. Furthermore, it allows for obtaining efficient representation of the data. The DNN model is implemented with an input layer, intermediate layers, and an output layer. Figure 8 presents the architecture of the proposed system based on the DNN model. At each layer, several settings are made to determine the number of neurons.

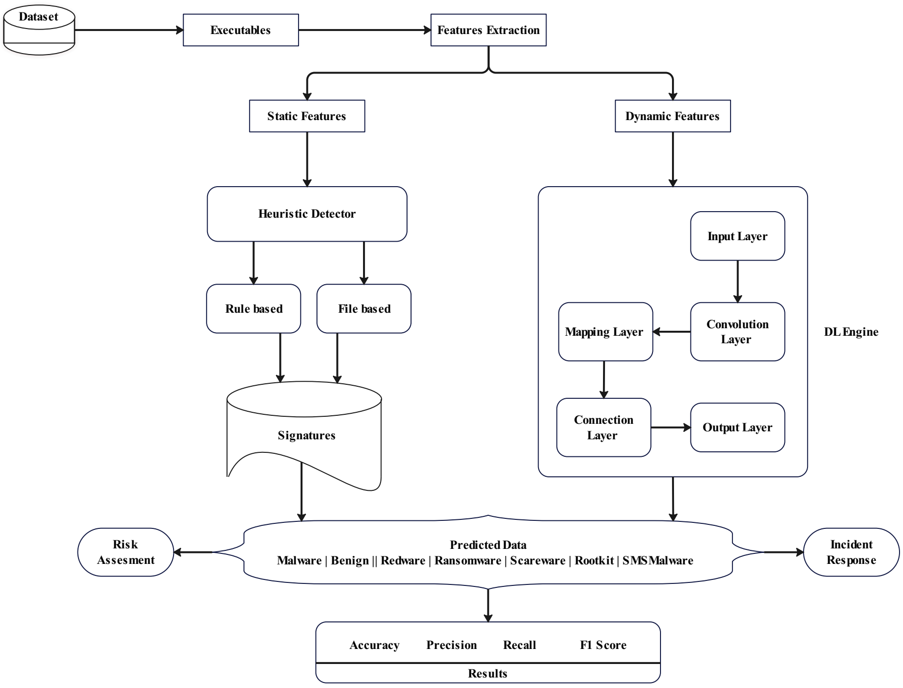


Figure 7. System architecture based on CNN for two classifications.

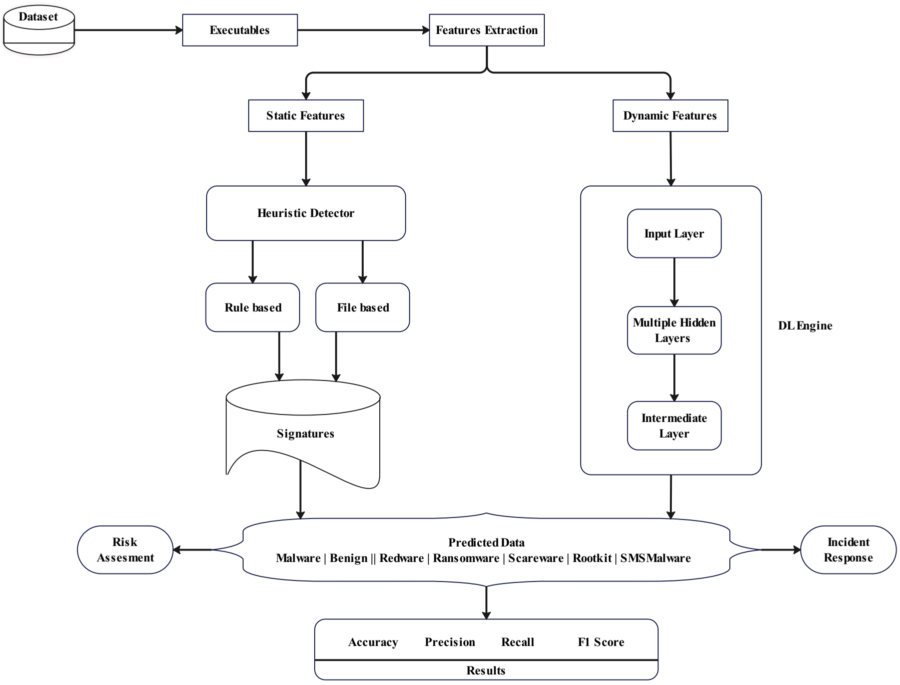


Figure 8. System architecture based on DNN for two classifications.

4.13. Malware Classification Based on ML

The machine learning algorithm in our study was carried out on the random forest (RF) classifier and the decision tree (DT) classifier. The architecture is presented in Figure 9.

Random forest and decision tree are machine learning algorithms that allow us to predict or classify. To do this, decision trees use the divide-and-conquer strategy. The process of building a decision tree is accomplished by selecting an attribute that will be a divisor of the dataset. Thus, it allows building explicit rules from multiple data based on the target variable in question.

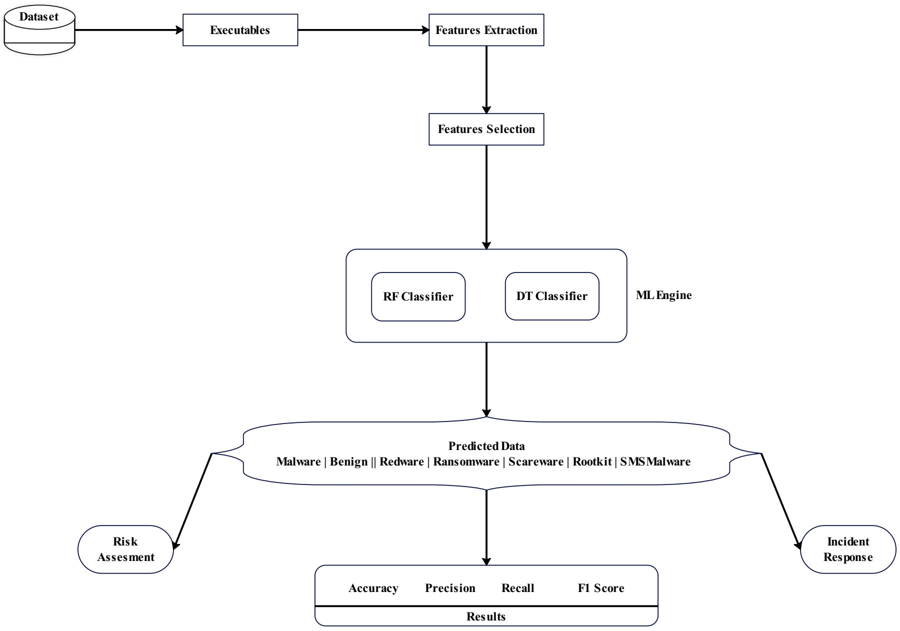
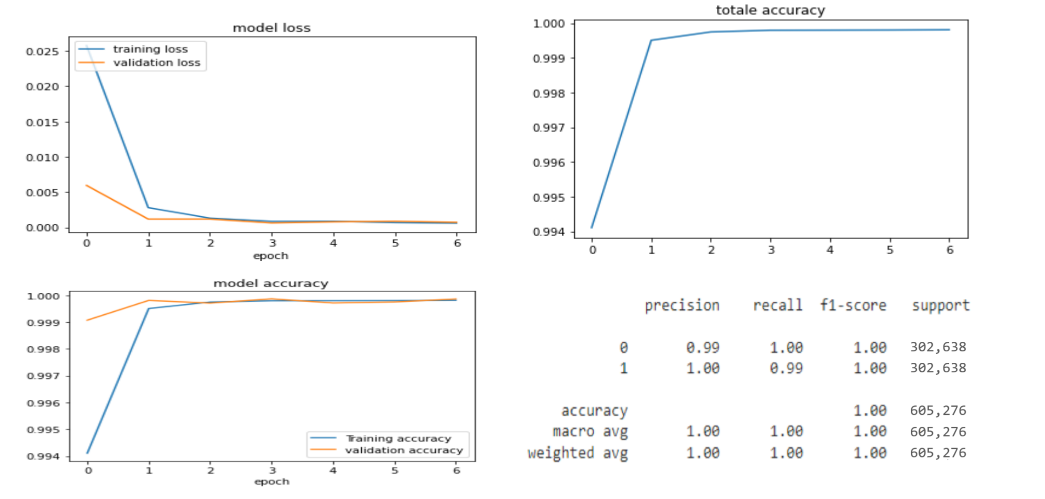


Figure 9. System architecture based on RF/DT for two classifications.

5. Experimental Results and Discussion

Our study aims to detect and classify modern malware with a negligible error rate. We implemented two deep learning algorithms (CNN and DNN) with heuristic detectors for static features and two traditional machine learning algorithms (RF and DT). These models were trained on two classification types (binary and multiple). We conducted several tests on all of the data to find the right hyperparameters. During development, when we had negligible test loss with some accuracy on precision, we tested these models on the test subset. The experimental results are presented in the following Tables 7–11:

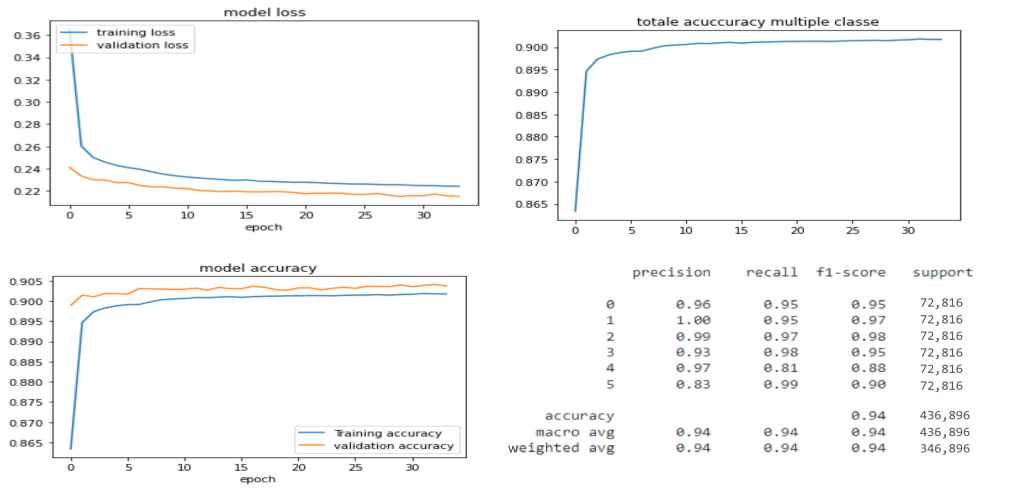
Table 7. DNN results for binary classification.

Loss/Accuracy and Total Accuracy


```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00   302,638
           1       1.00      0.99      1.00   302,638

    accuracy                           1.00   605,276
   macro avg       1.00      1.00      1.00   605,276
weighted avg       1.00      1.00      1.00   605,276
```

Table 8. DNN results for multi-class classification.

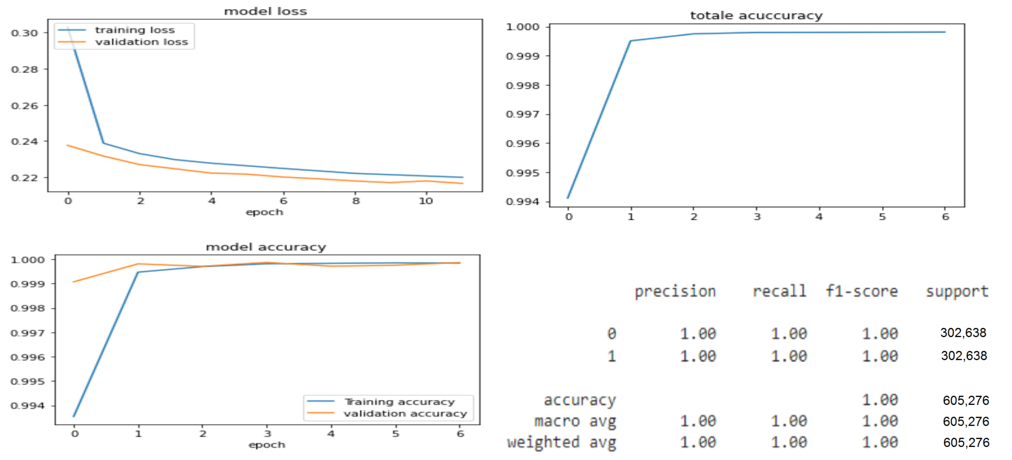
Loss/Accuracy and Total Accuracy



```
              precision    recall  f1-score   support

           0       0.96      0.95      0.95    72,816
           1       1.00      0.95      0.97    72,816
           2       0.99      0.97      0.98    72,816
           3       0.93      0.98      0.95    72,816
           4       0.97      0.81      0.88    72,816
           5       0.83      0.99      0.90    72,816

    accuracy                           0.94   436,896
   macro avg       0.94      0.94      0.94   436,896
weighted avg       0.94      0.94      0.94   346,896
```

Table 9. CNN results for binary classification.

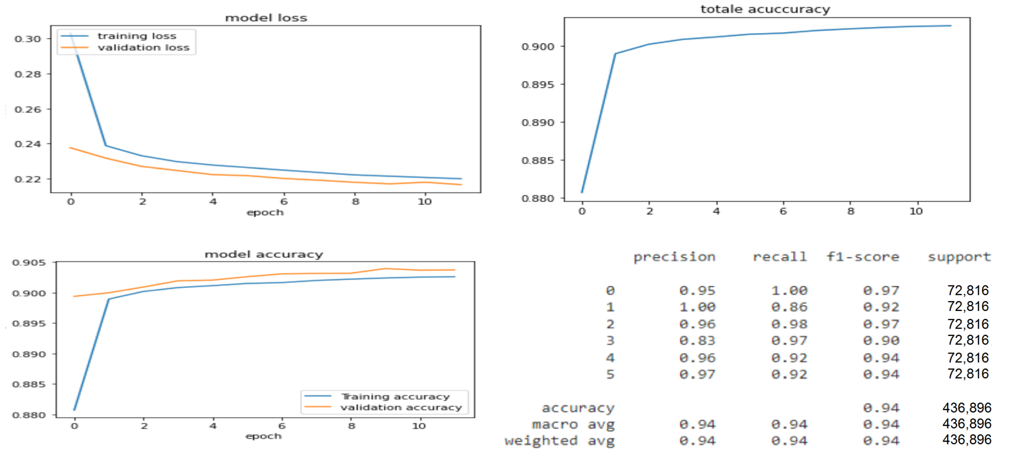
Loss/Accuracy and Total Accuracy



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    302,638
           1       1.00      1.00      1.00    302,638

    accuracy                           1.00    605,276
   macro avg       1.00      1.00      1.00    605,276
weighted avg       1.00      1.00      1.00    605,276
```

Table 10. CNN results for multi-class classification.

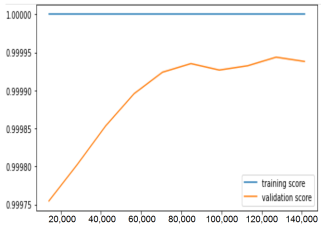
Loss/Accuracy and Total Accuracy

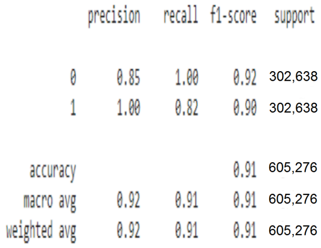


```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97    72,816
           1       1.00      0.86      0.92    72,816
           2       0.96      0.98      0.97    72,816
           3       0.83      0.97      0.90    72,816
           4       0.96      0.92      0.94    72,816
           5       0.97      0.92      0.94    72,816

    accuracy                           0.94    436,896
   macro avg       0.94      0.94      0.94    436,896
weighted avg       0.94      0.94      0.94    436,896
```

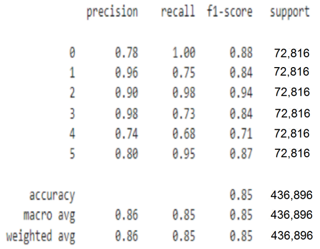
Table 11. RF and DT results for two classifications.

RF Binary and Multiple/DT Binary and Multiple

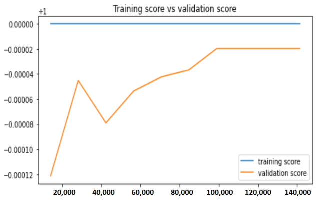


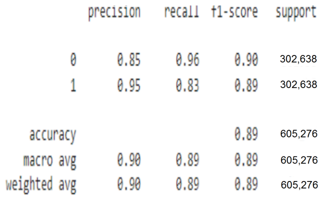


```
              precision    recall  f1-score   support

           0       0.85      1.00      0.92   302,638
           1       1.00      0.82      0.90   302,638

    accuracy                           0.91   605,276
   macro avg       0.92      0.91      0.91   605,276
weighted avg       0.92      0.91      0.91   605,276
```

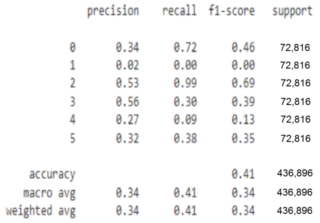


```
              precision    recall  f1-score   support

           0       0.78      1.00      0.88    72,816
           1       0.96      0.75      0.84    72,816
           2       0.90      0.98      0.94    72,816
           3       0.98      0.73      0.84    72,816
           4       0.74      0.68      0.71    72,816
           5       0.80      0.95      0.87    72,816

    accuracy                           0.85   436,896
   macro avg       0.86      0.85      0.85   436,896
weighted avg       0.86      0.85      0.85   436,896
```





```
              precision    recall  f1-score   support

           0       0.85      0.96      0.90   302,638
           1       0.95      0.83      0.89   302,638

    accuracy                           0.89   605,276
   macro avg       0.90      0.89      0.89   605,276
weighted avg       0.90      0.89      0.89   605,276
```



```
              precision    recall  f1-score   support

           0       0.34      0.72      0.46    72,816
           1       0.02      0.00      0.00    72,816
           2       0.53      0.99      0.69    72,816
           3       0.56      0.30      0.39    72,816
           4       0.27      0.09      0.13    72,816
           5       0.32      0.38      0.35    72,816

    accuracy                           0.41   436,896
   macro avg       0.34      0.41      0.34   436,896
weighted avg       0.34      0.41      0.34   436,896
```

5.1. Evaluation Results

In this section, we present the different evaluation metrics of the models we used. The results of precision, recall, F1 score, and the ratios of all metrics are presented in previous tables for the different classification models.

5.2. Accuracy

The accuracy identifies the total number of observations correctly identified with respect to the total number of observations. In our case, we denote the percentage of malware identified as malware among all examples predicted as malware, calculated by:

$$Accuracy = TP + TN/TP + TN + FP + FN \tag{1}$$

where the main aspects to consider when measuring the accuracy are:

- TP (true positive): malware that is correctly classified as malware;
- TN (true negative): normal that is correctly classified as benign;
- FP (false positive): normal that is incorrectly classified as malware;
- FN (false negative): malware that is incorrectly classified as benign.

5.3. Recall

Recall, also known as the true positive rate or sensitivity, represents the ability to detect all positive cases. In our case, we denote the percentage of malware identified as malware among all malware in the dataset. It is calculated by:

$$Recall = TP/TP + TN \tag{2}$$

5.4. F1-Score

The F-score measures the harmonic mean of precision and recall, which serves as a derived effectiveness measurement. It is calculated by:

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \tag{3}$$

5.5. False Positive Rate

This is the benign traffic that is classified as malware traffic. The FPR is calculated by:

$$FPR = FP/FP + TN \tag{4}$$

5.6. Confusion Matrix

The confusion matrix is the prediction result of our study. Figure 10 presents a confusion matrix for the DNN model with binary classification.

Our experimental results are compared to the following recent approaches in Table 12:

Table 12. Comparative study related to AI-based malware detection.

Source	Class	Model	Accuracy	Recall	F1-Score
Proposed	Adware Ransomware Scareware SMSMal Rootkit	Dynamic DL/Heuristic-based	1.00 0.99 0.93 0.97 0.97	0.95 0.98 0.98 0.92 0.99	0.97 0.98 0.95 0.94 0.99
[68]	Trojan	OWL/RE/SPARQL	0.92	0.91	-
[69]	Adware	SVM RF NB	0.96 0.99 0.94	-	-
[70]	Rootkits	RF	0.98	-	-
[71]	SMS Spam	SVM	0.98	-	-
[72]	AndroidMal	BiLSTM	0.98	-	-
[73]	Ransomware	ML	0.87	-	-
[74]	Scareware	DT	0.79	-	-
[75]	Android Mal	AdaBoost/SVM	0.96	-	-
[76]	Android Mal	LSTM CNN-LSTM EA	0.94 0.95 0.75	0.94 0.93 0.66	0.94 0.95 0.77
[41]	Adware	LSTM	0.95	-	-

The experimental results, which are benchmarked by testing our proposed approach with both binary and multi-class classification for the collected large malware datasets, show high accuracy with a deep learning model compared to the machine learning model. This indicates that advanced AI techniques, such as dynamic DL, can be an effective support base in cybersecurity to build intelligent and autonomous solutions as well as efficient mechanisms for modern malware detection and analysis.

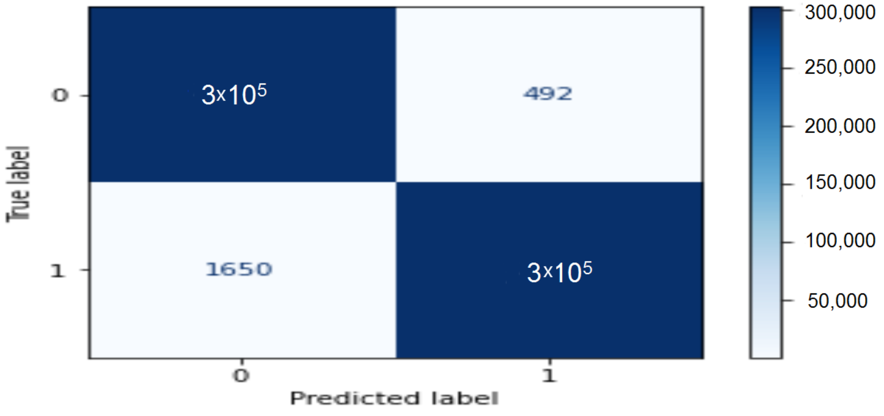


Figure 10. Confusion matrix for binary classification for the DNN model.

From the above comparisons, it appears that our method clearly outperforms existing approaches for malware detection and identification, predominantly for the classification of adware, ransomware, and scareware. The experimental results indicate that the dynamic DL model outperforms other proposed RF-based and RF-based machine learning methods. Furthermore, the combinations between behavior-based DNN and the heuristic-based approach and between behavior-based CNN and the heuristic-based approach provided better detection mechanisms than the use of ML and DL methods alone.

Moreover, the focus on dynamic features has a significant impact related to avoiding the obfuscation of modern malware evasion techniques. In addition, the deep analysis of threat and behavior indicators significantly increases the detection accuracy of malicious code with highly accurate vulnerability analysis.

For this reason, the experimental outcomes demonstrate that the dynamic DL methods combined with heuristic approaches are better in terms of detection rate, accuracy, and features selection, thus, they can significantly improve autonomously modern malware detection and mitigation.

Therefore, the proposed approach performed better than the existing malware detection and classification approaches. Our study provides new opportunities that can help lead to the more robust development of resilient systems against modern malware sophistication.

Moreover, since zero risk does not exist in cybersecurity, it is imperative to have at least an acceptable risk. To this end, by using fuzzy mathematics [77], we can improve risk assessment for cybersecurity issues, which will be the subject of further investigation.

6. Conclusions

Malware detection–analysis is an emerging topic in cybersecurity. Every year, many organizations and states are victims of malware attacks. A malware-based cyberattack can lead to devastating consequences, including financial loss, exfiltration of sensitive data, and cyber espionage. Malware scanners and conventional antiviral solutions cannot effectively meet protection needs. For this purpose, valuable malware examination helps to predict damage before it is produced and build innovative solutions to handle malware incidents. In this work, we propose a systematic approach to classify modern malware on an android dataset (CICAndMal2017) using dynamic deep learning-based methods combined with heuristic approaches to classify and detect five modern malware families—adware, Radware, rootkit, SMS malware, and ransomware. The performances of the proposed detection approaches were evaluated by taking into account the different evaluation measures. The experimental results show that the scores of the combination between behavior-based DNN and heuristic-based approach, and behavior-based CNN and heuristic approach have a better performance than the use of ML and DL methods alone. This vision of symmetry responds to the integration and intersection of two booming research areas: artificial intelligence and cybersecurity to promote and strengthen the security posture.

For future work, we plan to apply other data samples, to collect a lot of data taking into account all the characteristics and to use another type of learning (unsupervised or by reinforcement). We also advocate in-depth malware analysis through a variety of multi-sample approaches to effectively strengthen defense mechanisms and early response to mitigate breaches regardless of the platforms deployed.

Author Contributions: Conceptualization, A.D. and A.B.; methodology, A.D.; software, I.M.M.; validation, A.D. and A.B.; formal analysis, A.D. and A.B.; investigation, A.D.; resources, A.D. and A.B.; writing—original draft preparation, A.D.; writing—review and editing, A.D., A.B. and S.R.; visualization, A.D., A.B. and S.R.; supervision, A.D. and A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CICAndMal2017 free dataset from the Canadian Institute for Cybersecurity was downloaded from the following link: http://205.174.165.80/CICDataset/CICMalAnal2017/ (accessed on 31 January 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gobble, M.M. Digitalization, digitization, and innovation. *Res. Technol. Manag.* **2022**, *61*, 56–59. [CrossRef]
2. Jamsa, K. Cloud computing. In *Jones Bartlett Learning*; Springer: Berlin/Heidelberg, Germany, 2022.
3. Costa, B.; Bachiega, J., Jr.; de Carvalho, L.R.; Araujo, A.P. Orchestration in fog computing: A comprehensive survey. *ACM Comput. Surv.* **2022**, *55*, 1–34. [CrossRef]
4. Hartmann, M.; Hashmi, U.S.; Imran, A. Edge computing in smart health care systems: Review, challenges, and research directions. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3710. [CrossRef]
5. Gill, S.S.; Kumar, A.; Singh, H.; Singh, M.; Kaur, K.; Usman, M.; Buyya, R. Quantum computing: A taxonomy, systematic review and future directions. *Softw. Pract. Exp.* **2022**, *52*, 66–114. [CrossRef]
6. Sahani, A.; Sushree, B.B.P. The Emerging Role of the Internet of Things (Iot) in the Biomedical Industry. In *The Role of the Internet of Things (Iot) in Biomedical Engineering*; Apple Academic Press: Palm Bay, FL, USA, 2022; pp. 129–156.
7. Djenna, A.; Harous, S. Internet of things meet Internet of threats: New concern cyber security issues of critical cyber infrastructure. *Appl. Sci.* **2021**, *11*, 4580. [CrossRef]
8. Conti, M.; Dargahi, T.; Dehghantanha, A. *Cyber Threat Intelligence: Challenges and Opportunities*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 70, pp. 1–6.
9. Gandhi, R.; Sharma, A.; Mahoney, W.; Sousan, W.; Zhu, Q.; Laplante, P. Dimensions of cyberattacks: Cultural, social, economic, and political. *IEEE Technol. Soc. Mag.* **2011**, *30*, 28–38. [CrossRef]
10. Oz, H.; Aris, A.; Levi, A.; Uluagac, A.S. A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Comput. Surv.* **2022**, *54*, 1–37. [CrossRef]
11. Thangavel, K.; Plotnek, J.J.; Gardi, A.; Sabatini, R. Understanding and investigating adversary threats and countermeasures in the context of space cybersecurity. In Proceedings of the IEEE/AIAA 41st Digital Avionics Systems Conference, Portsmouth, NH, USA, 18–22 September 2022.
12. Sevis, K.N.; Seker, E. Cyber warfare: Terms, issues, laws and controversies. In Proceedings of the IEEE International Conference On Cyber Security And Protection Of Digital Services, London, UK, 13–14 June 2016.
13. Kaspersky Security Bulletin 2022. Statistics. Available online: https://securelist.com/ksb-2022-statistics/108129/ (accessed on 29 November 2022).
14. Harley Malware: New Attack on Android Devices. Available online: https://infosecwriteups.com/harley-malware-new-attack-on-android-devices-ae2c599c2217 (accessed on 5 June 2022).
15. Triada Trojan in WhatsApp Mod. Available online: https://securelist.com/triada-trojan-in-whatsapp-mod/103679/ (accessed on 11 June 2022).
16. Post-Infection Remediation. Available online: https://spycloud.com/ (accessed on 20 October 2022).
17. The SpyCloud Ransomware Defense Report 2022. Available online: https://spycloud.com/resource/ransomware-defense-report-2022/ (accessed on 30 November 2022).
18. The Ransomware Threat Landscape. Available online: https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/ransomware-threat-landscape-what-expect-2022 (accessed on 19 October 2022).
19. Malware. Available online: https://www.av-test.org/en/statistics/malware/ (accessed on 16 October 2022).
20. Yin, H.L.; Fu, Y.; Li, C.L.; Weng, C.X.; Li, B.H.; Gu, J.; Lu, Y.S.; Huang, S.; Chen, Z.B. Experimental quantum secure network with digital signatures and encryption. *Natl. Sci. Rev.* **2022**, nwac228. [CrossRef]
21. Gu, J.; Cao, X.Y.; Fu, Y.; He, Z.W.; Yin, Z.J.; Yin, H.L.; Chen, Z.B. Experimental measurement-device-independent type quantum key distribution with flawed and correlated sources. *Sci. Bull.* **2022**, *67*, 2167–2175. [CrossRef]
22. Will Quantum Computing Define The Future of AI? Available online: https://analyticsindiamag.com/will-quantum-computing-define-the-future-of-ai/ (accessed on 8 February 2023).
23. Zhou, M.G.; Cao, X.Y.; Lu, Y.S.; Wang, Y.; Bao, Y.; Jia, Z.Y.; Fu, Y.; Yin, H.L.; Chen, Z.B. Experimental quantum advantage with quantum coupon collector. *Res. Sci. Partn. J.* **2022**, *2022*, 9798679. [CrossRef]
24. SonicWall Cyber Threat Report. Available online: https://theblockchaintest.com/uploads/resources/SonicWall%20-%20Cyber%20Threat%20Report%20-%202022%20Feb.pdf (accessed on 3 December 2022).
25. Debnath, P.; Mohiuddine, S.A. *Soft Computing Techniques in Engineering, Health, Mathematical and Social Sciences*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2021; pp. 1–232.
26. Langner, R. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Secur. Priv.* **2011**, *9*, 49–51. [CrossRef]
27. The Flame: Questions and Answers. Available online: https://securelist.com/the-flame-questions-and-answers/34344/ (accessed on 10 September 2022).
28. The Epic Turla (Snake/Uroburos) Attacks. Available online: https://www.kaspersky.com/resource-center/threats/epic-turla-snake-malware-attacks (accessed on 3 September 2022).

29. Mohurle, S.; Patil, M. A brief study of WannaCry threat: Ransomware attack 2017. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 1938–1940.
30. Microsoft Security Bulletin MS17-010-Critical. Available online: https://learn.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010 (accessed on 27 September 2022).
31. Aidan, J.S.; Garg, U. Advanced Petya ransomware and mitigation strategies. In Proceedings of the IEEE First International Conference on Secure Cyber Computing and Communication, London, UK, 15–17 December 2018.
32. Greenberg, A. The untold story of NotPetya, the most devastating cyberattack in history. *Wired* **2018**, *22*, 1–14.
33. Kraszewski, K. SamSam and the silent battle of Atlanta. In Proceedings of the IEEE 11th International Conference on Cyber Conflict, Tallinn, Estonia, 28–31 May 2019.
34. Davidson, R. The fight against malware as a service. *Netw. Secur.* **2021**, *8*, 7–11. [CrossRef]
35. Djenna, A.; Saidouni, D.E.; Abada, W. A pragmatic cybersecurity strategies for combating IoT cyberattacks. In Proceedings of the IEEE International Symposium on Networks, Computers and Communications, Montreal, QC, Canada, 20–22 October 2020.
36. Wireshark. Available online: https://www.wireshark.org/ (accessed on 31 October 2022).
37. TCP DUMP. Available online: https://www.tcpdump.org/ (accessed on 31 October 2022).
38. Bernardi, L.; Mavridis, T.; Estevez, P. 150 successful machine learning models: 6 lessons learned at booking.com. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–8 August 2019.
39. Jerlin, M.A.; Marimuthu, K. A new malware detection system using machine learning techniques for API call sequences. *J. Appl. Secur. Res.* **2018**, *13*, 45–62. [CrossRef]
40. Zhu, D.; Jin, H.; Yang, Y.; Wu, D.; Chen, W. DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In Proceedings of the IEEE Symposium on Computers and Communications, Heraklion, Greece, 3–6 July 2017.
41. Catak, F.O.; Yazı, A.F.; Elezaj, O.; Ahmed, J. Deep learning based Sequential model for malware analysis using Windows exe API Calls. *PeerJ Comput. Sci.* **2020**, *6*, e285. [CrossRef]
42. Milosevic, N.; Huang, J. Deep learning guided Android malware and anomaly detection. *arXiv* **2019**, arXiv:1910.10660.
43. HaddadPajouh, H.; Dehghantanha, A.; Khayami, R.; Choo, K.K.R. A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Gener. Comput. Syst.* **2018**, *85*, 88–96. [CrossRef]
44. Karbab, E.B.; Debbabi, M. MalDy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports. *Digit. Investig.* **2019**, *28*, S77–S87. [CrossRef]
45. Huang, Y.P.; Basanta, H. Bird image retrieval and recognition using a deep learning platform. *IEEE Access* **2019**, *7*, 66980–66989. [CrossRef]
46. Zhu, H.J.; Gu, W.; Wang, L.M.; Xu, Z.C.; Sheng, V.S. Android malware detection based on multi-head squeeze and excitation residual network. *Expert Syst. Appl.* **2023**, *212*, 118705. [CrossRef]
47. Seraj, S.; Khodambashi, S.; Pavlidis, M.; Polatidis, N. MVDroid: An Android malicious VPN detector using neural networks. *Res. Sq.* **2022**, 1–14. [CrossRef]
48. Sasidharan, S.K.; Thomas, C. ProDroid—An Android malware detection framework based on profile hidden Markov model. *Pervasive Mob. Comput.* **2021**, *72*, 101336. [CrossRef]
49. Lee, Y.; Wang, X.; Liao, X.; Wang, X. Understanding Illicit UI in iOS Apps Through Hidden UI Analysis. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 2390–2402. [CrossRef]
50. Yang, R.; Chen, X.; Xu, H.; Cheng, Y.; Xiong, C.; Ruan, L.; Chen, Y. Ratscope: recording and reconstructing missing rat semantic behaviors for forensic analysis on windows. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1621–1638. [CrossRef]
51. Azmoodeh, A.; Dehghantanha, A.; Conti, M.; Choo, K.K.R. Detecting crypto-ransomware in IoT networks based on energy consumption footprint. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *9*, 1141–1152. [CrossRef]
52. Rahul; Kedia, P.; Sarangi, S.; Monika. Analysis of machine learning models for malware detection. *J. Discret. Math. Sci. Cryptogr.* **2020**, *23*, 395–407. [CrossRef]
53. Nguyen, G.L.; Dumba, B.; Ngo, Q.D.; Le, H.V.; Nguyen, T.N. A collaborative approach to early detection of IoT Botnet. *Comput. Electr. Eng.* **2022**, *97*, 107525. [CrossRef]
54. Cheng, Q.; Wu, C.; Zhou, H.; Kong, D.; Zhang, D.; Xing, J.; Ruan, W. Machine learning based malicious payload identification in software-defined networking. *J. Netw. Comput. Appl.* **2021**, *192*, 103186. [CrossRef]
55. Gopinath, M.; Sethuraman, S.C. A comprehensive survey on deep learning based malware detection techniques. *Comput. Sci. Rev.* **2023**, *47*, 100529.
56. Anwar, A.; Halabi, T.; Zulkernine, M. Scalable Collaborative Intrusion Detection in Autonomous Vehicular Networks: A hierarchical framework based on game theory. *Internet Things* **2022**, *20*, 100631. [CrossRef]
57. Han, W.; Xue, J.; Wang, Y.; Zhang, F.; Gao, X. APTMalInsight: Identify and cognize APT malware based on system call information and ontology knowledge framework. *Inf. Sci.* **2021**, *546*, 633–664. [CrossRef]
58. Almashhadani, A.O.; Carlin, D.; Kaiiali, M.; Sezer, S. MFMCNS: A multi-feature and multi-classifier network-based system for ransomworm detection. *Comput. Secur.* **2022**, *121*, 102860. [CrossRef]
59. Beaman, C.; Barkworth, A.; Akande, T.D.; Hakak, S.; Khan, M.K. Ransomware: Recent advances, analysis, challenges and future research directions. *Comput. Secur.* **2021**, *111*, 102490. [CrossRef]
60. Khammas, B.M. Ransomware detection using random forest technique. *ICT Express* **2020**, *6*, 325–331. [CrossRef]

61. Fernando, D.W.; Komninos, N. FeSA: Feature selection architecture for ransomware detection under concept drift. *Comput. Secur.* **2022**, *116*, 102659. [CrossRef]
62. Qiu, J.; Han, Q.L.; Luo, W.; Pan, L.; Nepal, S.; Zhang, J.; Xiang, Y. Cyber Code Intelligence for Android Malware Detection. *IEEE Trans. Cybern.* **2022**, *53*, 617–627. [CrossRef]
63. Qiao, Y.; Zhang, W.; Tian, Z.; Yang, L.T.; Liu, Y.; Alazab, M. Adversarial ELF Malware Detection Method Using Model Interpretation. *IEEE Trans. Ind. Inform.* **2022**, *19*, 605–615. [CrossRef]
64. Xu, J.; Li, Y.; Deng, R.H.; Xu, K. Sdac: A slow-aging solution for android malware detection using semantic distance based api clustering. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1149–1163. [CrossRef]
65. Fan, M.; Liu, J.; Luo, X.; Chen, K.; Tian, Z.; Zheng, Q.; Liu, T. Android malware familial classification and representative sample selection via frequent subgraph analysis. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1890–1905. [CrossRef]
66. Canadian Institute for Cybersecurity. Available online: https://www.unb.ca/cic/ (accessed on 1 May 2021).
67. Lashkari, A.H.; Kadir, A.F.A.; Taheri, L.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark android malware datasets and classification. In Proceedings of the IEEE International Carnahan Conference on Security Technology, Montreal, QC, Canada, 22–25 October 2018.
68. Aboshady, D.; Ghannam, N.; Elsayed, E.; Diab, L. The Malware Detection Approach in the Design of Mobile Applications. *Symmetry* **2022**, *14*, 839. [CrossRef]
69. Pavithra, J.; Selvakumara Samy, S. A Comparative Study on Detection of Malware and Benign on the Internet Using Machine Learning Classifiers. *Math. Probl. Eng.* **2022**, *2022*, 4893390. [CrossRef]
70. Wang, X.; Zhang, J.; Zhang, A.; Ren, J. TKRD: Trusted kernel rootkit detection for cybersecurity of VMs based on machine learning and memory forensic analysis. *Math. Biosci. Eng.* **2019**, *16*, 2650–2667. [CrossRef] [PubMed]
71. Jain, T.; Garg, P.; Chalil, N.; Sinha, A.; Verma, V.K.; Gupta, R. SMS spam classification using machine learning techniques. In Proceedings of the 12th IEEE International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 27–28 January 2022.
72. Bayazit, E.C.; Sahingoz, O.K.; Dogan, B. A Deep Learning Based Android Malware Detection System with Static Analysis. In Proceedings of the IEEE International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022.
73. Khan, F.; Ncube, C.; Ramasamy, L.K.; Kadry, S.; Nam, Y. A digital DNA sequencing engine for ransomware detection using machine learning. *IEEE Access* **2020**, *8*, 119710–119719. [CrossRef]
74. Bagui, S.; Brock, H. Machine Learning for Android Scareware Detection. *J. Inf. Technol. Res.* **2022**, *15*, 1–15. [CrossRef]
75. Urooj, B.; Shah, M.A.; Maple, C.; Abbasi, M.K.; Riasat, S. Malware detection: A framework for reverse engineered android applications through machine learning algorithms. *IEEE Access* **2022**, *10*, 89031–89050. [CrossRef]
76. Alkahtani, H.; Aldhyani, T.H. Artificial intelligence algorithms for malware detection in android-operated mobile devices. *Sensors* **2022**, *22*, 2268. [CrossRef]
77. Mordeson J.N.; Nair, P.S. *Fuzzy Mathematics An Introduction for Engineers and Scientists*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 1–314.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.