IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 February 2023, accepted 10 April 2023, date of publication 19 April 2023, date of current version 28 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3268535

SURVEY

The Age of Ransomware: A Survey on the Evolution, Taxonomy, and Research Directions

SALWA RAZAULLA[1], CLAUDE FACHKHA[1], CHRISTINE MARKARIAN[1], (Member, IEEE), AMJAD GAWANMEH[1], (Senior Member, IEEE), WATHIQ MANSOOR[1], (Senior Member, IEEE), BENJAMIN C. M. FUNG[2], (Senior Member, IEEE), AND CHADI ASSI[3], (Fellow, IEEE)

[1]College of Engineering and IT, University of Dubai, Dubai, United Arab Emirates
[2]School of Information Studies, McGill University, Montreal, QC H3A 1X1, Canada
[3]Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

Corresponding author: Claude Fachkha (cfachkha@ud.ac.ae)

This work was supported by the University of Dubai under the Internal Research Fund.

ABSTRACT The proliferation of ransomware has become a significant threat to cybersecurity in recent years, causing significant financial, reputational, and operational damage to individuals and organizations. This paper aims to provide a comprehensive overview of the evolution of ransomware, its taxonomy, and its state-of-the-art research contributions. We begin by tracing the origins of ransomware and its evolution over time, highlighting the key milestones and major trends. Next, we propose a taxonomy of ransomware that categorizes different types of ransomware based on their characteristics and behavior. Subsequently, we review the existing research over several years in regard to detection, prevention, mitigation, and prediction techniques. Our extensive analysis, based on more than 150 references, has revealed that significant research, specifically 72.8%, has focused on detecting ransomware. However, a lack of emphasis has been placed on predicting ransomware. Additionally, of the studies focused on ransomware detection, a significant portion, 70%, have utilized Machine Learning methods. This study uncovers a range of shortcomings in research pertaining to real-time protection and identifying zero-day ransomware, and two issues specific to Machine Learning models. Adversarial machine learning exploitation and concept drift have been identified as under-researched areas in the field. This survey is a constructive roadmap for researchers interested in ransomware research matters.

INDEX TERMS Ransomware, malware analysis, machine learning, deep learning, cyber attacks, adversarial machine learning.

I. INTRODUCTION

The widespread and hugely publicized WannaCry outbreak of 2017 put the spotlight back on ransomware [1]. This attack not only demonstrated how potentially dangerous ransomware could be but also exemplified the extent of its profitability. The main motive of the WannaCry attack was not monetary gains but chaos and panic. While the ransom demand was only a mere $300, the financial damage went well beyond that of the ransom itself, estimated to be around $4 billion. Since then a myriad of ransomware attacks and variants have emerged. The increase in recent cyber-attacks is also greatly attributed to the COVID-19 pandemic [2]. As companies shifted to a remote work paradigm, employees became more susceptible to phishing emails, thereby introducing security gaps in the organization's defense against cyber-attacks. But what makes this type of malware so distinctive? Ransomware is a malicious piece of software that is designed to deny or minimize users' access to their files, operating system, or device and demands a ransom payment in order to regain access [3]. In general, ransomware is classified into two broad categories, namely locker ransomware, which encrypts files essential for basic computer functions, and cryptographic ransomware which encrypts user's

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Moinul Hossain.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

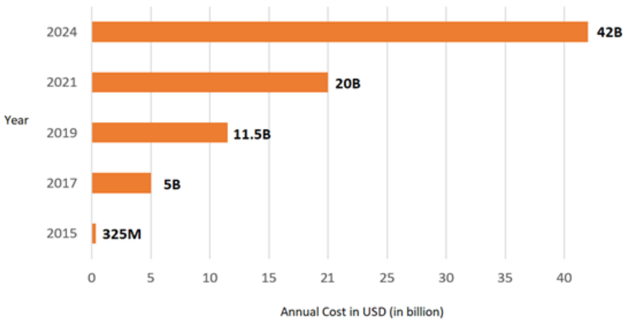

FIGURE 1. Global Damage Caused by Ransomware Attacks.

sensitive files [4]. This infamous malware has targeted a wide range of targets, including individual users, business enterprises, government entities, and hospitals to name a few. Ransomware made its initial appearance in the year 1989 and while it has been around for over three decades now, its variants have grown progressively advanced in their encryption methods, capability to spread quickly, evade detection, and compel victims into paying the ransom [5]. Ransomware has quickly risen up the ranks-becoming one of the most prominent and ubiquitous types of malware. Cybersecurity Ventures predicts the global ransomware damage to exceed $265 billion by 2031, with a new attack every 2 seconds [6]. Figure 1 depicts the total damage caused by ransomware attacks globally between the years 2015 to 2024 [7].

Although there have been numerous studies that have surveyed and summarized different solutions for defending against ransomware, these surveys have focused on specific components of ransomware research. However, no study has provided a holistic understanding of the evolution, a comprehensive taxonomy of ransomware, defense research of ransomware across multiple platforms, such as desktop, mobile devices, IoT, and ICS systems, and different goals of ransomware defense. This is a crucial research gap, as understanding the complete picture is becoming increasingly important in countering this rapidly growing threat. Motivated by the above-mentioned concerns, we provide a comprehensive overview of ransomware and the recent research contributions for ransomware security. To the best of our knowledge, this is the first study to provide an in-depth research comparison and classification that includes details such as the overlap of these different research works, and a breakdown of different Machine Learning techniques used, just to name a few.

In essence, the key contributions of this paper are summarized as follows:

- Provide a survey on ransomware by studying their roots and principal components, history and events since 1989, and research trends for the past 7 years.
- Create a taxonomy for ransomware research ideas that categorizes them and shows where they overlap in regard to analysis techniques.
- Determine research gaps, then offer ideas and suggestions for further research such as the ones that are related to offensive and adversarial machine learning approaches.

The remainder of this paper is organized as follows:

Section II provides an overview of ransomware evolution and highlights the key stages of the cyber kill chain. In addition, it shows the most common infection vectors for ransomware. Section III gives the related work. A survey of the recent advances in ransomware security research is presented in Section IV. Section V provides a discussion on the open research problems that need to be addressed in future ransomware defense research. Finally, Section VI concludes our work with our findings of the current research contributions for countermeasures against ransomware.

II. BACKGROUND

The following section presents an overview of ransomware and emphasizes the focus of our survey by (1) discussing the different types of ransomware; (2) outlining various stages of the ransomware kill chain; (3) providing some common infection vectors; and finally (4) comprehensively exploring ransomware evolution.

A. TYPES OF RANSOMWARE

There have been two major categories of ransomware namely, *crypto* and *locker* ransomware. More recently other types are gaining popularity among attackers. We list out four of the more traditional variants of ransomware [5]:

1. Crypto - As the most common type of ransomware, crypto-ransomware aims to encrypt data important to victims, such as documents, pictures, and videos, but not to interfere with basic computer functions. Crypto-ransomware typically allows victims to view the list of encrypted files and use the system, but they are unable to access the actual files that are encrypted. Data encrypted by crypto-ransomware using current techniques such as AES and RSA is often irrecoverable, as these encryption methods are almost irreversible if implemented correctly [8].

2. Locker - This type of ransomware locks the victims out of their systems. In the majority of cases, victims of Locker ransomware are typically only allowed to view the lock screen or a screen with ransom payment instructions. These types of ransomware attacks are often relatively easy to resolve and can be dealt with by rebooting the computer in safe mode or running an on-demand virus scanner [9].

3. Scareware - This ransomware-type tricks users into downloading or buying malicious or sometimes useless software by displaying startling messages, often done using pop-up ads. Users who take the bait inadvertently install ransomware on their devices. This type of ransomware does not necessarily pose a real threat to its victim [10].

4. Leakware – also known as Doxware, is a new and potent form of ransomware that threatens to make users' data public unless the ransom is paid. The damage caused is irreversible as anyone can access the data once it is open to the public [11].

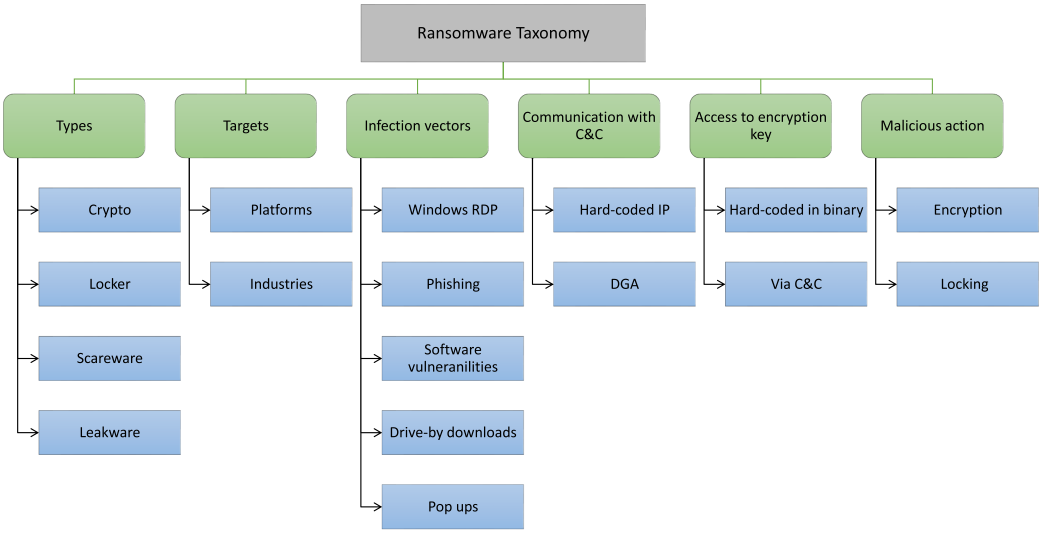

FIGURE 2. **Taxonomy of Ransomware.**

Banks and organizations that handle confidential or sensitive information are particularly at risk of being targeted by this type of attack.

Figure 2 illustrates the classification of ransomware according to its types, attack vectors, communication methods with Command and Control servers, and the malicious actions it carries out, providing a comprehensive taxonomy of this threat.

Ransomware as a service (RaaS) - RaaS is a ransomware distribution model similar to Software-as-a-Service (SaaS) model, where attackers lease out ransomware attacks to other cybercriminals. The services provided by this model can include the compiled ransomware, ransomware customization tools, and infrastructure for maintaining the ransomware, instructions among others. Such type of services enables even those criminals who lack the skills or time to develop their own ransomware variants to quickly and inexpensively launch attacks [12]. These ransomware kits are easily available on the dark web and include many payment models such as one-off ransomware purchase, on a commission basis, or a monthly subscription. The widespread adoption of the ransomware-as-a-service (RaaS) model has contributed to the steady growth of ransomware attacks in recent years [13].

A number of notorious Ransomware-as-a-Service variants exist, including:

1. *Ryuk* – attributed to the hacker group WIZARD SPIDER, is one of the most successful and costly variants of ransomware [14]. It is estimated to have generated approximately $150 million in profits by the end of 2020.

2. *Maze* – The concept of double extortion was first introduced by this specific variant of ransomware, where cybercriminals steal sensitive data and demand payment in exchange for not publicly releasing it. Although Maze has discontinued its operations, similar variants such as Egregor continue to thrive, operating through the Ransomware as a Service (RaaS) model.

3. *Lockbit* – This variant emerged in late 2019 and has been around since. The hallmark of Lockbit is its ability to swiftly encrypt the systems of giant corporations, reducing the time available for defenders to detect and remove the malware before harm is inflicted.

4. *REvil* - Also known as Sodinokibi, was the malware behind one of the biggest ransom demands on record, a staggering $10 million. This specific ransomware variant is spread through a multitude of methods, and it has been reported that its affiliates utilize unpatched Citrix and Pulse Secure VPNs as a means of infiltrating and infecting systems.

B. RANSOMWARE KILL CHAIN

Ransomware attacks typically follow six primary stages: distribution, infection, staging, scanning, encryption, and payment [15].

- Distribution - The initial phase of the attack involves spreading the malware to the targeted device. Some of the ways attackers accomplish this are with phishing emails, exploit kits, malicious websites, or vulnerabilities in the connection or user system.
- *Infection* – At this phase, the ransomware is installed on the machine and begins the infection process.
- *Staging* – During the staging phase, ransomware embeds itself in the system, establishes persistence to survive beyond a reboot, and begins communicating with the outside world. This frequently entails uploading victims' information to a domain that has recently been registered or to an IP address.
- *Scanning* – During this step, the malware scans both the local computers and network resources, in search of data that can be encrypted, including network drives and cloud storage accounts like Box.com and Dropbox [16].

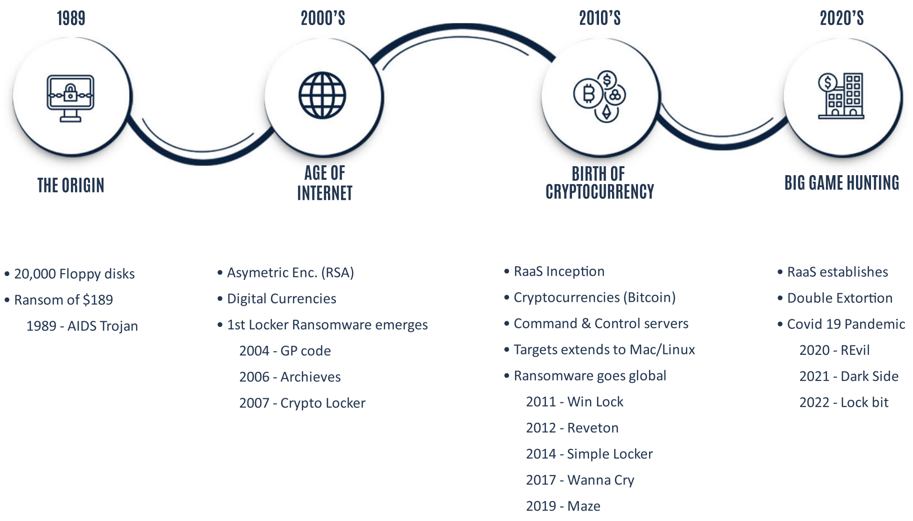

FIGURE 3. Evolution of Ransomware.

- *Encryption*- Once ransomware locates important user files, it starts encrypting using the encryption keys hard-coded in its binary or the ones acquired from a C&C server.
- *Payment* – The final step is displaying a ransom note on the screen and waiting to collect the ransom.

C. INFECTION VECTORS

The most common ransomware attack vectors are:

- *Phishing* - Phishing still dominates as the most used ransomware infection vector [17]. A typical attack attempt begins when a user receives a malicious email that contains links, attachments, or both with instructions. The users are tricked into clicking or opening the attachment as the email appears to be from a known contact. Common file formats such as pdf, doc, and jpg are used to ensure recipients run the executable file.
- *Remote Desktop Protocol (RDP)* - RDP is the second most popular attack vector after phishing attacks. RDP is considered an ideal attack vector because attackers keep finding new and lesser-known vulnerabilities daily. For instance, in 2020 alone, 25 new vulnerabilities were discovered in RDP clients.
- *Software vulnerabilities* – Vulnerabilities take the third spot among the most common infection methods. In some cases, when software is not properly updated or patched, attackers can access networks without having to harvest credentials.
- *Web pages* - Ransomware can also be found in a seemingly legitimate or compromised website, hidden in web scripts on these sites. What makes this a perfect infection vector is that users believe they are visiting a trusted site. Ransomware is automatically downloaded on a user's machine when a user visits that site.
- *Pop-ups* - Another common web-based attack vector is pop-ups, which trick users into clicking them by appearing genuine and posing as legitimate sources. Ransomware is either automatically downloaded on the victim's computer, or directed to a new window with malicious links.

D. EVOLUTION OF RANSOMWARE

This section delves deeper into some of the significant ransomware variants that emerged throughout each decade since its emergence, providing a comprehensive understanding of the progression of this threat. Figure 3 depicts the progression of ransomware, beginning with its inception in 1989, through the era of rapid internet expansion, and culminating in the present-day utilization of Ransomware-as-a-Service models and double extortion tactics by attackers. This illustration provides a comprehensive overview of the evolution of this threat.

Ransomware first emerged late in 1989 when a professor, called Dr. Popp, distributed 20,000 virus-infected floppy disks to people at the international AIDS conference. Once it was loaded onto a system, the virus began hiding directories, locking files, and required a payment of $189 for the restoration of access to the affected data [18]. Ironically, Dr. Popp was neither a computer scientist nor a programmer but a biologist. He was eventually arrested and charged with 10 counts of blackmail and causing damage through the distribution of what is now referred to as the ''AIDS Trojan''. Eventually, it was determined that he was incapable of standing trial due to psychological reasons. Ransomware took a long hiatus of 15 years since its emergence in 1989 [14]. The next time it appeared was with the advent of digital and crypto-currencies allowing for a more elegant form of payment. The re-emergence of ransomware was also driven by the widespread adoption of the internet and email as daily tools for communication and business. At the early stage of the internet era, two of the most significant ransomware attacks were GPCode and Archievus. These attacks were different from today's ransomware, as the attackers requested a low ransom because they preferred targeting a high volume of victims, rather than targeting a smaller number of high-value victims. GPCode, which surfaced in 2004, used two infection vectors to attack victims, namely phishing emails and malicious website links. By 2006, Archievus marked a shift in the evolution of ransomware as it was the first strain to use Rivest-Shamir-Adleman (RSA) encryption [19]. This evolution of encryption technology showed how cyber criminals had been adapting to the changing landscape of cyber security. The year 2007 saw the emergence of the first locker ransomware variants that locked victims' machines and prevented them from using their computers' basic functions. WinLock led this era of ransomware. It operates by taking control of the victim's screen and displaying explicit images, forcing the victim to pay a ransom via paid SMS to regain access to their computer [20]. This type of malware represented a unique and particularly aggressive form of ransomware that caused widespread concern among computer users and security experts alike.

A couple of years later, analysts learned of 2013's most malicious malware threat called CryptoLocker. By December of 2013, this potent form of ransomware had impacted roughly 250,000 Windows-based computers. It was also during this time that security researchers learned that cybercriminals were not only targeting professionals but also home-based internet users. The primary source of infection during this year seemed to be phishing emails that contain malicious attachments. In mid-2012, a password-stealing malware named Reveton ransomware, also referred to as Win23/Reveton, the FBI Virus, or the Police Trojan, made its appearance [14]. This later evolved into ransomware that exploited hundreds of thousands of dollars from its victims every month. It achieved this by posing as law enforcement agencies to deceive victims and coerce them into paying a "fine" or facing the consequences of being arrested. 2014 marked a significant milestone in the evolution of ransomware when SimpleLocker made its debut, becoming the first strain to target Android devices and encrypt images, documents, and videos stored on SD cards [14]. This new strain expanded the potential targets to include a wider range of victims and opened the door to a whole new set of attacks. Probably the most notorious malware infection of all time was the infamous WannaCry of 2017, a crypto-ransomware worm that attacks Windows PCs. This is still actively used by cyber attackers today. Maze ransomware first surfaced in May 2019 and has been highly active since December 2019. The malware not only encrypts data but also exfiltrates the targeted data, threatening to release it publicly unless the victims pay a ransom. This type of attack can have severe consequences for businesses, as it uses double extortion with regular ransomware actions. This makes it a particularly concerning threat for organizations. Furthermore, the information on this type of malware is constantly evolving and new attack methods are being developed by the cybercriminals behind it.

One of the worst threats that 2020 saw was in the form of Egregor ransomware. Egregor ransomware is a highly sophisticated form of malware that has gained notoriety for its brutal double-extortion tactics. Despite its destructive capabilities, little is known about this ransomware as it employs various anti-analysis techniques such as payload encryption and code obfuscation to evade detection and analysis. Egregor is believed to have links to the now-defunct Maze ransomware. Conti ransomware is particularly destructive due to its rapid data encryption speed and ability to spread to other systems. The Conti group often uses phishing attacks to install TrickBot and BazarLoader Trojans, granting them remote access to infected machines. After encrypting the data, Conti follows a two-step extortion process. DarkSide, which initially appeared in mid-2020, was responsible for the attack on the Colonial Pipeline, termed the most devastating cyber-attack of 2021. The group is known to only attack organizations that can pay a huge amount of ransom, rather than targeting governments, non-profit organizations, and hospitals [21]. In 2022, the highest number of cyber-attacks was from among the newer variants that also employed double extortion tactics. Lockbit ransomware was able to quickly make its mark in the Raas space due to its ability to upgrade its attack techniques frequently.

E. MALWARE ANALYSIS

Malware analysis is the process of examining a malware specimen in order to determine its characteristics, origin, behavior, purpose, and potential impact. The outcome of such analysis helps security teams to rapidly detect, prevent, and mitigate potential threats. In most cases, malware analysis is the preliminary step that researchers have used in ransomware defense. There are three types of malware analysis:

1. *Static analysis* - This type of analysis inspects the binary file without executing the malicious program. It can reveal important information about the malware's command and control infrastructure, targets, and persistence mechanisms. Features such as hashes, strings, opcodes, byte sequences, etc are extracted and analyzed to determine if the file is malicious. Various network analyzers and disassembler tools inspect the malware without having to run the binary code. Some commonly used static analysis tools include PeStudio, ExeInfo PE, HxD, CyberChef, and IDA Pro. The principal benefit of static analysis is its speed. However, it is not effective against sophisticated malware strains that use code obfuscation techniques to evade detection [9].

2. *Dynamic analysis* - This involves executing a sample of the malicious software in a controlled environment that is separated from the host system [24]. This allows researchers to observe the behavior of the malware without putting the host system at risk. Researchers can use various tools such as a virtual machine, a sandbox, for example, Cuckoo sandbox,

TABLE 1. Comparison of Related Works.

Publication	Platform				Years	Evolution of Ransomware	Research goals				
	Desktop	Mobile	IoT	ICS			Detection	Classification	Prevention	Mitigation	Prediction
[18]	-	-	-	-	2016 - 2020	✗	✓	✗	✗	✗	✗
[22]	-	-	-	-	2016 - 2020	✗	✓	✗	✗	✗	✗
[9]	-	-	-	-	2015 - 2020	✗	✓	✗	✓	✓	✗
[14]	✓	✓	✓	-	1999 - 2020	✓	✓	✗	✓	✓	✗
[23]	-	-	-	-	2015 - 2018	✗	✓	✗	✗	✗	✗
This survey	✓	✓	✓	✓	2016 - 2022	✓	✓	✓	✓	✓	✓

✓ Yes ✗ No - Information not provided

or a debugger to monitor the activity of the ransomware, including the files and processes it creates, the network connections it establishes, and any other actions it takes. Dynamic analysis can also be used to determine the ransom payment mechanism and the encryption algorithm used by the ransomware. Additionally, it can be used to identify the mechanism used by ransomware to propagate itself, such as exploiting vulnerabilities or using phishing emails.

3. *Hybrid analysis* - It is a method of analyzing malware that combines both static and dynamic analysis techniques. Combining both static and dynamic analysis provides a more comprehensive view of the malware and its capabilities. Hybrid analysis is a powerful technique that can be used to analyze a wide range of malware, including ransomware, trojans, and other types of malicious software.

III. RELATED WORK

In this section, we present two categories of related survey contributions, namely malware security studies and ransomware security studies. Table 1 gives a comparison of related works and this survey.

Ye et al. [25] surveyed intelligent malware detection approaches which included the broad stages of feature extraction and classification or clustering. The paper also highlighted the challenges of malware detection using data mining techniques and estimate the future trends of malware development. Souri and Hosseini [26] summarized the challenges related to detection approaches for malware using data mining techniques and provided a comparison of these approaches with respect to the classification method, size of dataset used, analysis approach, and accuracy. s. The pros and cons of various data mining models were discussed in terms of their evaluation method, proficiency, and overall effectiveness. Faruki et al. [27] surveyed the techniques employed for Android security. In particular, they covered the malware detection methods as well as the stealthy techniques used by attackers. This review offers a comprehensive examination of the strengths and weaknesses of established research methodologies in the realm of Android security. It aims to provide researchers and practitioners with a foundation for proposing innovative approaches to analyze and combat these threats.

Another study that surveyed Android malware was by Tam et al. [28]. The study presents a survey of the most effective Android malware analysis and detection techniques and their ability to keep up with evolving malware. It evaluates the systems by methodology, malware statistics, and evasion techniques, as well as industry solutions, and suggests future research paths. Ucci et al. [29] focused on the use of machine learning in malware analysis for Windows environments specifically for Portable Executable. It categorizes papers based on their goals, and the machine learning techniques employed. The survey also highlights challenges, including those related to datasets, and future possibilities for advancement.

Bello et al. [18] presented a taxonomy of ransomware research works that only used intelligent machine learning algorithms. Their survey considered papers between 2016 and 2020. Finally, the authors have outlined some of the possibilities of future directions and challenges in the application of deep learning techniques for ransomware defense. The survey conducted by Fernando et al. [22] explored contributions that made use of machine learning and deep learning algorithms for detecting ransomware. The potential impact of the evolving nature of ransomware on these works was also determined through experiments. They further explored the future advancements of ransomware and its evolution in the years to come. Beaman et al. [9] highlighted the recent cutting-edge approaches for detecting and preventing ransomware attacks. Finally, the authors created a ransomware prototype, named AESthetics, which successfully evaded detection by eight widely used antivirus programs. Oz et al. [14] adopted a distinctive methodology of analyzing research works on ransomware, specifically with regard to the diversity of the platforms that are targeted. The authors have covered works on mobile devices, IoT platforms as well as PCs/workstations. However, the authors have not discussed studies on prevention against ransomware. Berrueta et al. [23] examined the various detection techniques developed by the researchers for ransomware. The proposed algorithms were compared and classified based on the features obtained from ransomware behavior and the decision procedures used to reach a classification conclusion. It offers a comprehensive overview of detection algorithms and a comparison of the results.

Our survey, which aligns with the second group of surveys, focuses on research for ransomware analysis and defense. The primary distinction between the research in [9], [22] and ours is the scope of the research goals. While their focus is on detecting and preventing ransomware, our work covers a broader range of topics, including detection,

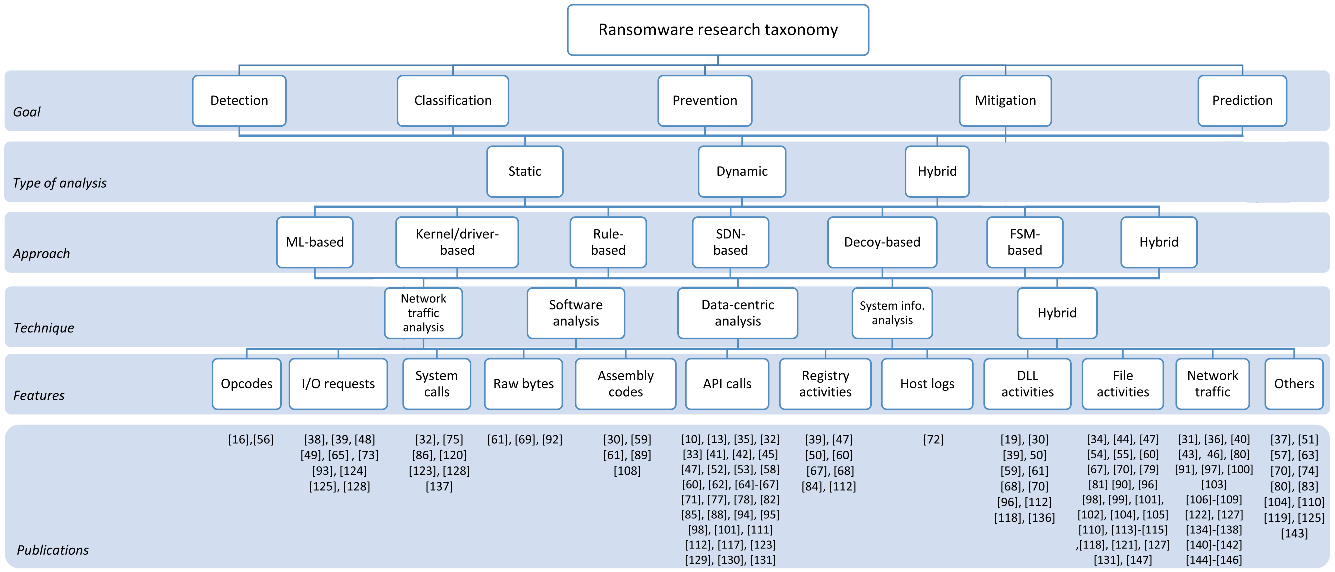

FIGURE 4. **Ransomware Research Taxonomy.**

classification, mitigation, prevention, and prediction of ransomware. We have recently added a new category called prediction, which includes all contributions related to predicting ransomware. The work in [18] focuses on studies on detection based on machine learning algorithms, whereas our work offers a comprehensive analysis of research that employs a variety of other techniques such as honey pots, Software-Defined Networks (SDN) in addition to machine learning. Additionally, to the best of our knowledge, none of the existing survey papers have addressed adversarial machine learning for ransomware, which we briefly cover in Section V.

IV. RANSOMWARE RESEARCH TAXONOMY

In this section, we examine the latest research studies focused on combating ransomware. We have taken into account the most recent works, dating from 2016 and beyond. These studies aim to achieve a variety of research goals, which we have broadly categorized into five areas: detection, classification, prevention, mitigation, and prediction.

Figure 5 shows the trend of selected 125 ransomware research publications between 2016 and 2022 that we considered for this paper, with the highest point of publication activity being recorded in 2018. Among the noteworthy contributions are the detection of zero-day ransomware attacks [30], [31], [32], [33], [34], [35], [36], [37], [38], the identification of ransomware at an early stage, both before it commenced file encryption and examined its surroundings, and finally the transformation of user devices into analysis-like environments, effectively obscuring them from evasive malware [39]. Figure 4 depicts our taxonomy of ransomware research contributions according to the various analysis types, approaches, and features that were used. We have categorized the approaches and techniques into categories. Rule-based approaches are those that detect ransomware using a set of rules that are built based on the extracted features, whereas decoy-based approaches deploy decoys or honeypots as bait to catch and identify ransomware attacks. Hybrid approaches involve a combination of two or more other approaches. The techniques level in the figure refers to the type or category of data that is analyzed by the research papers. These include software analysis, which involves examining the binary code of the sample, while data-centric analysis focuses on analyzing the users' data. System information analysis includes analysis of system data, such as configurations, running processes, kernel activities, and other system-related information. Finally, the features level highlights the distinctive features utilized by the research studies in their analysis. The majority of the research studies have used API calls because they provide a rich source of information about software behavior, are reliable, are used in dynamic analysis, and are easily extracted and analyzed. The network traffic features encompass a variety of data points, including IP addresses, TCP headers, packet size, protocols used, payload content, domain information, and other network-related parameters.

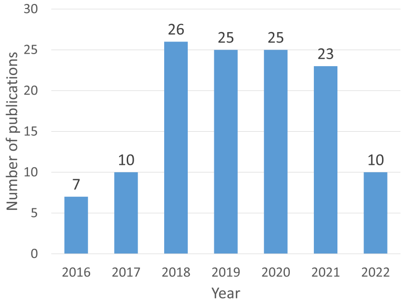

FIGURE 5. **Publications Trend Year-wise (solely on this survey).**

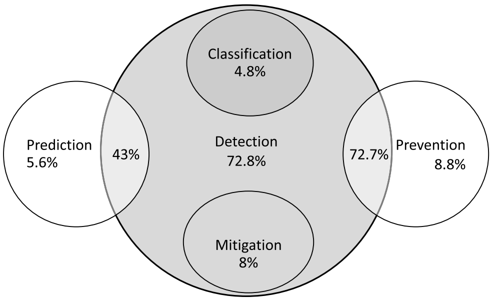

FIGURE 6. Type of Analysis Distribution in Literature Studies.

The file activities features comprise a range of actions performed on files such as creation, deletion, and modification. Additionally, it takes into account various characteristics of the files themselves, such as file similarity, entropy, size, and more. Finally, there were a smaller number of features that did not fall under any of the pre-existing categories, which we grouped into the category of ''Others''. These features include information on Bitcoin transactions, patterns of power and energy consumption by applications, and images and texts from XML layout files on Android devices, to name a few examples. Figure 6 presents the distribution of studies that address various goals such as detection, classification, prevention, mitigation, and prediction, as well as the overlap between these goals. The majority of the studies (72.8%) were focused on detection. The focus on classification, prevention, mitigation, and prediction was significantly lower, at 4.8%, 8.8%, 8%, and 5.6% respectively. The figure also shows that all the studies that focused on the classification and mitigation of ransomware also included ransomware detection, while 72.7% of the preventive studies and 43% of the prediction papers also performed the detection of ransomware.

A. DETECTION

Ransomware detection is the process of identifying the presence of ransomware on a system or network. Researchers have employed a variety of methods to detect ransomware, including machine learning algorithms, and honeypots. In this section, we discuss these contributions. Figure 7 illustrates the various analysis types utilized by researchers in detecting ransomware.

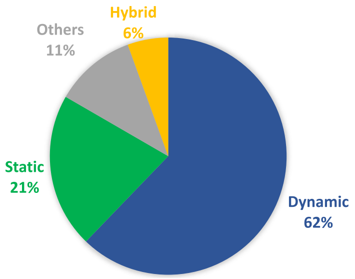

FIGURE 7. Type of Analysis Distribution in Literature Studies.

1) MACHINE LEARNING-BASED DETECTION

The main advantage of machine learning, including deep learning, is that it can learn from historical data and identify patterns that indicate the presence of ransomware. Additionally, machine learning algorithms can be trained to detect new and unknown variants of ransomware, which is important given the constantly evolving nature of this threat. Tables 2 and 3 present a summary of the publications that focus on using machine learning for ransomware detection. This section presents a review of the literature on the detection of ransomware using machine learning. The review is organized by the platforms that have been targeted by the research, including desktop platforms, mobile platforms, and other platforms such as IoT systems, SCADA systems, and cloud platforms among others.

a: DESKTOP PLATFORMS

In order to curb the issue with static analysis, Chen et al. [42] implemented dynamic analysis to derive a set of API call flow graph (CFG) to build the feature vector space. They have employed simple logistic among other data mining algorithms for detecting malicious software from benign software. Vinayakumar et al. [45] implemented two models namely, shallow and deep networks by leveraging API invocations. Dynamic analysis of seven ransomware families was performed to collect a set of API calls that were used as features to the proposed multi-layer perceptron (MLP). Hasan and Rahman [47] proposed a hybrid ransomware detection approach that combines both static and dynamic analysis. The experiments covered samples from all recent ransomware families including WannaCry. The results revealed that the proposed hybrid approach can detect ransomware with higher accuracy results. Baek et al. [48] proposed a new set of lightweight behavioral functions for ransomware override patterns. In fact, the function relies on monitoring block I/O request headers, not only payloads. For complete and instant recovery, the authors also utilized SSD's delayed erase feature, which is unique to NAND flash. They implemented a working prototype called SSD-Insider. In the experiment, which contains around ten real and in-house ransomware programs, the authors found that SSD-Insider detects and recovers ransomware within seconds. Cohen and Nissim [50] proposed a trusted ransomware detection methodology for virtual servers in organizational private clouds. The authors showed that their approach can detect anomalous virtual machine states, in addition to known and unknown ransomware. They also showed that the latter can detect a new malware type, called RAT, known to attack virtual machines of organizations. Takeuchi et al. [52] employ supervised learning and train the model using a set of API calls to detect unseen ransomware samples. Dynamic analysis is performed to extract the features from the malicious execution logs.

Al-rimy et al. [53] introduced a new model for early detection that overcomes the problem of not having complete data available. It is based on two new techniques,

TABLE 2. Summary of Machine Learning-based Ransomware Detection Contributions.

Publication	Features	Machine learning algorithms		Accuracy	Dataset		
		ML classifier	DL technique		Source	# Ransomware families	# Samples
[10]	API calls	Ensemble DT			RISS	10	582 R 942 B
[15]	-	LinR		87.9%	-	11	582 R 942 B
[16]	Opcode sequence		CNN	89.5%	Virus Total	8	100 B
[40]	Network traffic,System activities		DNN	98.3%	ISoT dataset*	-	-
[41]	API calls, app permissions, metadata, images, network, memory		DNN	98.1%	R-PackDroid, HellDroid, Contagio	8	1928 R 2500 B
[35]	System API packages	RF		-	HellDroid, Virus Total	-	2045 R 4098 B
[42]	API calls flow graph(CFG)	RF, SVM, SL, NB		98.2%	Virus Share		83 R 85 B
[43]	Network traffic	SVM	NN			-	-
[44]	App Permissions, Package names, URLs and use of obfuscation Number of files in an APK, its size, Number of permissions Activities and services, the average class size, Total number of packages and the number of classes contained	SVM, SGD, DT JRip, FURIA, LAC, RIDOR		93.7%	Contagio Mobile Virus Total	-	-
[45]	API calls, frequencies	SVM	MLP	98.0%	Open Malware, Contagio Malware Dump Malwr, theZoo aka Malware DB4 VirusTotal, VirusShare	7	755 R 219 B
[46]	Network traffic	RF			-	-	-
[47]	Function Length Frequency(FLF), Printable string info(PSI) API functions, Registry key oprations, File operations	SVM		97.1%	Virus Share	21	360 R 460 B
[48]	I/O request headers	DT		100.0%	-	10	-
[49]	IRPs	RF, NB, LogR, DT		96.6%	Virus Total, Open Malware VXVault, Zelster, Malc0de	14	261 R + B
[50]	Windows Registry, and lists of running processes, services Dynamic-link libraries (DLLs) in use	RF		96.6%	-	5	-
[51]	Power/energy consumption patterns	kNN, SVM, RF	NN	83.7%	Virus Total	-	6 R 12 B
[52]	API calls	SVM		97.5%	Hybrid Analysis	-	276 R 312 B
[53]	API calls	SVM, LogR, RF, DT, AdaBoost, kNN	MLP		Virus share	15	8152 R 1K B
[54]	File similarity	K-means clustering (DM)		-	Hybrid Analysis, Malshare	1	112 R
[55]	Entropy of files	kNN, LinR, LogR, RidgeR, LassoR, LinSVC, DT Ensemble, kernel trick,RF, SVM, GB	MLP	100%	-	-	-
[56]	Opcode sequence	RF,DT,kNN GB-DT,NB		99.3%		8	1787 R 100 B
[57]	Email's header, Email's domain Sender's IP address, subject, Message, attachment	-	-	-	-	-	-
[58]	API packages calls	RF, SVM, DT J48, NB		97.0%	HelDroid project, Virus Total Ransomware proper project, Koodous	-	500 R 500 B
[59]	DLLs, function call, Assembly instruction	NB, RF, LogR SVM, DT		98.6%	Virus Total, theZoo	-	292 R
[60]	API calls, registry key actions, File system actions File extensions file names, Directory actions Application folders, Control panel settings, C&C server	NB, DT		96.3%	Virus Total Malware blacklist	8	10K R 500 B
[61]	Raw binaries, DLL libraries Assembly codes, Function calls	BN, LogR, SVM, DT J48 RF, AdaboostM1 with J48		97.8%	Virus Total, Virus Share, theZoo	13	178 R
[62]	API calls		RNN, LSTM	93.0%		-	-
[63]	Time domain Hardware Performance Counter(HPC) data		LSTM		-	1	-
[64]	API sequences	RF, LogR, NB, SGD, kNN, SVM		98.7%	Virus Total	-	1K R 300 B
[65]	APIs IRPs		LSTM			-	-
[66]	API calls		CNN	95.9%	-	-	1K R 1K B
[33]	API calls	RF			VirusTotal, theZoo	-	904 R 942 B
[67]	API calls, registry key operations File system operations, Strings, file extensions Directory operations, dropped file extensions	RF Regularized LogR	DNN	97.3%	-	-	2507 R 3886 B
[68]	DLL activities, File system actiities Registry activities	DT J48, RF, Bagging, MLP		99.4%	Virus Total	3	1624 R 220 B
[69]	Raw bytes	RF		97.7%		3	840 R 840 B
[70]	Name, MD5, bitcoin address, DLL characteristics, ImageBase, FileAlignment	GTB, NB, DT, AdaBoost		100.0%	Virus Total	8	-
[71]	API calls	LR, SVM, DT, RF, kNN, AdaBoost	MLP	94.0%	Virus Share	15	39378 R 16057 B
[36]	System logs, Network logs	kNN, DT, RF		98.5%	VirusShare, Malwaredb.Malekal	-	1054 R
[72]	Host logs		BiLSTM	99.9%	-	17	-
[73]	IRP logs		ANN	99.8%	-	18	272 R
[74]	PE header: DOS Header, PE Signature COFF Header, Optional Header, Section Header		CNN	93.3%	Virus share	4	1K R 1K B
[75]	System call	DT, kNN, LogR, RF, SVM			Virus Total Virus Share	14	1354 R 1358 B
[76]	-	RF, DT, NS Sequential Minimal Optimization		98.3%	RansImDS-API&Permissions (Koodus, RansomProper Project, Virus Total)	-	500 R 500 B
[77]	API calls	kNN, RF, SVM		99.2%	Virus Share, theZoo, Malwaretips Hybrid analysis, Malware sample sources	12	58 R 66 B

namely, incremental bagging, and enhanced semi-random subspace selection, incorporated into a detection model, that is ensemble-based. The results showed that the proposed techniques can achieve better accuracy results in comparison to other approaches. PhAttApp, a phishing-detecting app, was suggested by Lam and Kettani [57]. By providing a variety of capabilities to identify and stop ransomware transmission through phishing channels, this program lowers the chance of ransomware infection. The tool leverages header information to extract insights and take action accordingly. Poudyal et al. [59] proposed a framework that employed Natural Language Processing(NLP), machine learning techniques, and reverse engineering to classify executable files. Static analysis of the samples and N-gram and TF-IDF generated the feature vector. For faster processing of the large feature set, Apache Spark was used. Kok et al. [10] suggested a pre-encryption detection method (PEDA) with two phases. Their contribution mainly covered the first one, which is based on an untrustworthy program's Windows application programming interface (API) that would be recorded and subjected to the learning algorithm's analysis. Furthermore, this phase leverage API pattern recognition that allows the

TABLE 3. ***Contd.*** **Summary of Machine Learning-based Ransomware Detection Contributions.**

Publication	Features	Machine learning algorithms		Accuracy	Dataset		
		ML classifier	DL technique		Source	# Ransomware families	# Samples
[78]	API calls	RF, NB, kNN	ANN, LSTM	94.9%	-	21	19499 R 4500 B
[79]	No. of bytes read No. of bytes written Control Commands	DT TE	NN	99.9%	Hybrid analysis Malware Traffic Analysis	33	70 R
[80]	Valuable set of RAM File system Network features	Log R, kNN SVM, RF, NB		99.0%	-	10	50 R 50 B
[81]	Access priviledges: Read/Write/Execute/Copy	XGBoost,DT TE,GBT,NB SVM,NN		9628.0%	Malware Bazaar The Zoo	70	117 R 354 B
[31]	Network traffic		CNN		Malware Traffic Analysis.net PacketTotal Online Malware Analysis Sandbox	7	8175 R 20K B
[82]	API calls	LogR, kNN, RF DT, AD, SVM		98.63%	VirusShare Maltrieve, VirusTotal	16	1050 R 450 B
[83]	BGP update messages parameters	GBDT, GRU	CNN, RNN, LSTM	84.3%	-	1	-
[84]	Registry data	DT, Bayes NB, RF, SVM, and JRip					-
[85]	API calls App permissions		ELM	98%	HelDroid project RansomProper project Virus Total, Koodous	-	500 R 500 B
[86]	System calls	RF, J48, NB		98.3%	Virus Total	-	400 B
[87]	-		ELM	99.7%	VXHaven Kaggle Ransomware dataset [Homayoun paper]	-	-
[88]	API calls	GBDT		98.7%	Virus Total, Virus Share	8	1803 R 4008 B
[89]	Assembly instructions, Network traffic		DNN, CNN, LSTM	98.70%	Virus Total	-	561 R 447 B
[32]	API call, system calls	RF			Virus Share, theZoo	11	995 R 942 B
[90]	Entropy metrics	SVM linear SVM kernel trick(poly)		85.17% 92%	-	4	22 B
[91]	Network traffic	DT, RF, GB, NB, SVM		99.8%	-	6	-
[30]	DLL, function calls assembly levels	LogR, SVM,RF, DT J48, AdaBoost with RF, AdaBoost with J48	NN	99.7%	Virus Total		6 R 550 B
[92]	Raw bytes	RF, SVM , NB		98.3%	VirusTotal, ShieldFS Dataset [93]	-	600 R 600 B
[94]	Permissions and API calls	SVM			VirusTotal, Ransomware Proper, Koodous	-	500 R 9653 B
[95]	API calls	SVM, LogR, DT, kNN, RF, AdaBoost	MLP	97.08%	Virus Share	15	39378 R 16057 B
[96]	# of process start, # of process end # of DLL image loads, # of DLL image unloads # of file reads, # of file writes, # of threads start, # of thread end	SVM	NLP-based deep learning model called BERT	99.5%	VirusTotal, MalwareBazaar Live malware repository (github) Malwares samples(github)	67	292 R 54 B
[37]	Sensor data, CPU load	LogR, DT, RF, LinR, kNN, NB, One-V-One One-V-Rest, Extra tree, K-means	MLP		-	-	-

learning algorithm to assess whether the suspicious program was crypto-ransomware or not. Zuhair and Selamat [60] have presented a real-time detection system for Windows-based systems. Their system implements a hybrid algorithm which is a combination of two machine learning algorithms, namely Naíve Bayes and Decision Tree, to detect zero-day ransomware variants. Agrawal et al. [62] incorporated Attended Recent Inputs (ARI) by integrating attention in learning from malware sequences. The authors spotted repeating patterns possibly representing repeated encryption and proposed an LSTM variation of ARI that leverages these patterns to detect ransomware. Azman et al. [84] presented a framework using registry data as features and various machine learning algorithms such as SVM, Decision Tree, Random Forest, JRIP, and Naive Bayes for classification. Experiments using these algorithms are conducted on registry data affected by ransomware. Bae et al. [64] suggested a new ransomware detection approach that not only distinguishes between malicious software and benign files but also between ransomware and malware. The authors proved the effectiveness of their approach through extensive experiments. Qin et al. [66] presented a dynamic ransomware detection method that analyzes API calls of unknown executables to assess whether these files are malicious or not. This is achieved through the utilization of a text classification pooling layer from TextCNN and chunk-based max-pooling.Dynamic Ransomware Detector based on identifying whether an API call sequence can be regarded as a sentence in the language or not. This is based on using the text classification pooling layer of TextCNN and chunk-based max-pooling. The authors do not clarify the correlation between API calling sequence and ransom behavior, which might be a main factor in the accuracy of the detection process.

Kok et al. [33] intended to evaluate various learning algorithms that are adopted in ransomware detection, Authors proposed a set of conventional metrics and unconventional metrics for this purpose. Six new metrics were proposed for this purpose. The authors also proposed the use of various indices in order to compare different learning algorithms. Hwang et al. [67] built a two-stage mixed ransomware detection model based on Markov and Random Forest models. The approach focuses initially on Windows API call sequence patterns and leverages the Markov model to extract ransomware characteristics. The proposed mixed detection model can achieve high accuracy with low errors.

Homayoun et al. [68] proposed a method to collect activity logs for various types of ransomware. Then, they suggested a technique utilizing Sequential Pattern Mining to uncover Maximal Frequent Patterns (MFP) from the recorded activities across diverse ransomware families. These MFPs serve as potential features for classification through the use of various machine learning algorithms. Based on the identified distinctive frequent patterns within different ransomware families, authors were able to identify and detect ransomware in the dataset with high accuracy. The work is not validated with other types of ransomware that are not from the collected set. Motivated by the simplicity of static malware analytics, Khammas [69] proposed an approach for detecting ransomware. In fact, the technique was based on extracting features from raw bytes via data mining (e.g., frequent pattern mining). Similarly, Zhang et al. [16] claimed to be the first who classified ransomware through a self-attention system on opcode sequences. In fact, they detect fingerprinting ransomware using a static analysis framework using deep learning with N-gram opcodes. Jahromi et al. [87] developed a modified version of the Extreme Learning Machine (ELM) which performs faster and is simpler than Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) methods. This two-hidden-layered model uses the dependencies between malware sequence elements. Al-Rimy et al. [71] implements an annotated Term Frequency-Inverse Document Frequency (aTF-IDF) method to extract relevant API calls more reliably. Different ML models were employed including linear regression, and support vector machines to detect ransomware for Windows systems. Using dynamic analysis to capture network logs, Moussaileb et al. [36] try to distinguish ransomware network traffic from that of benign traffic. The authors have also evaluated the ransom note and the encrypted file to determine the time of their detection method. Roy and Chen [72] developed DeepRan, a deep learning-based ransomware detection system. Considerable volumes of host logs are collected from bare metal servers and fed to a bi-directional Long Short Term Memory (LSTM) in order to provide early detection for ransomware activity. During behavior analysis of ransomware samples, Ayub et al. [73] extracted discriminating IPR logs. An artificial neural network was built and fed these features to derive meaningful patterns in the logs. Khan et al. [15] proposed DNAact-Ran, a detection model that uses machine learning approaches to classify samples. A new feature set of ''genome rules'' based on DNA sequences is presented for high-reliability detection.

Manavi and Hamzeh [74] introduced a new technique for detecting ransomware which, unlike previously known approaches, can be used without running the given program. Instead, it extracts features from the PE headers of the executable files. Their technique is based on using Convolutional Neural Networks for classification and is shown to be 93.33% accurate. Ahmed et al. [75] proposed a ransomware detection approach that uses supervised machine learning techniques and is non-signature-based. They introduced the so-called Enhanced Maximum-Relevance and Minimum-Redundancy (EmRmR) filter method for understanding ransomware behavior with fewer call traces. They showed that their approach is effective with high accuracy and a low false-positive rate. Almousa et al. [77] designed a new approach for ransomware detection based on Application Programming Interface and Machine Learning. Their study used dynamic analysis on the Windows platform and sandbox analysis for sampling. Their results showed that the proposed approach can be used with existing multi-layer security solutions and 99.18% accuracy for Windows platforms.

Nguyen and Lee [88] proposed a method for ransomware detection by examining API calls extracted during dynamic analysis of executables in a virtual environment. ML is used for training, detecting, and classifying normal software as well as different types of ransomware. Hsu et al. [90] presented a detection tool that focuses on analyzing files rather than executable programs. They analyzed 22 formats of encrypted files and using Support Vector Machine extracted detailed features to tell whether a file is encrypted or not. Aljubory and Khammas [92] employed machine learning algorithms to classify and detect ransomware families. Static analysis was used to extract the raw bytes and build the feature space. Irrelevant features were removed using the gain ratio technique. Ahmed et al. [96] designed a new ransomware detection tool, called, Peeler. The latter monitors the kernel events of a given system and detects ransomware attacking the system. Their experiments included 43 ransomware families and showed 99% detection with 0.58% false-positive rate. Kok et al. [32] proposed a so-called Pre-encryption Detection Algorithm (PEDA) that can detect malware that locks files using encryption algorithms, known as crypto-ransomware. PEDA offers two levels of detection. The first is a detection before the activation of the ransomware, using signature comparison with known crypto-ransomware. The second is the detection of crypto-ransomware based on a pre-encryption Application Program Interface (API), using a learning algorithm. Molina et al. [78] designed a novel approach for ransomware classification that makes use of so-called paranoia activities, a series of API calls executed by ransomware to find a suitable execution environment. The authors fingerprinted the paranoia activities associated with more than 3K samples of recent and well-known ransomware families. Furthermore, they showed that their approach can produce a 94.92% classification accuracy. Singh et al. [81] proposed a detection method based on the examination of access privileges in the process memory. By utilizing process memory, the key functions of ransomware are more accurately and easily detected. Ahmed et al. [82] developed a new technique for early ransomware detection, referred to as Weighted minimum Redundancy maximum Relevance (WmRmR). The latter is capable of assessing important features from the relevant sets. Their experiments showed that WmRmR is effective for early detection with a low false-positive rate and complexity.

Figure 8 illustrates the different platforms utilized in the 125 literature studies for ransomware detection that we have considered. Given that the Windows OS is the primary target of ransomware attacks, the majority of the publications in the field of malware detection have focused on this platform. The 'others' category encompasses a range of platforms such as Cloud, IoT, SCADA, etc.

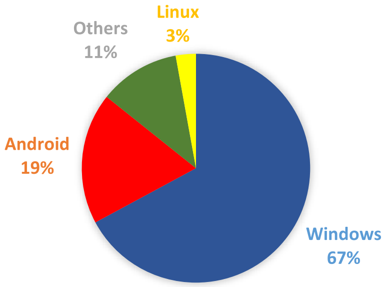

FIGURE 8. Platform Distribution in Literature Studies.

b: MOBILE PLATFORMS

Gharib and Ghorbani [41] proposed a detection framework composed of two layers, a dynamic analysis layer, and a static analysis layer. The authors showed that the latter can detect ransomware in the initial stages before infection occurs, with a high precision rate and a 1.5% false negative rate. Maiorca et al. [35] proposed R-PackDroid, which is a system dedicated to detecting Android ransomware via machine learning. In fact, this system leverages API packages to achieve its goal with high accuracy. A ransomware detector for mobile devices, called Greateatlon, was proposed by Zheng et al. [44]. The system uses static analysis to detect malicious use of APIs that are called by ransomware for encryption. In order to investigate mobile-based ransomware, Alsoghyer and Almomani [58] shed the light on both static and dynamic analysis of ransomware detection. In particular, an API-based ransomware detection system (API-RDS) was proposed for inferring android applications and corresponding ransomware families. Faris et al. [85] proposed an Android ransomware detection framework using a combination of metaheuristic and machine learning methods. Raw API call sequences and permissions are used to capture ransomware patterns and build the framework. Abdullah et al. [86] performed dynamic analysis to capture the system calls. These features were fed to Random Forest among other machine learning algorithms to detect and classify the samples. Almomani et al. [76] introduced a new detection approach for Android ransomware, based on machine learning techniques. Their study was made on Android Version 11, API level 30, and used a number of predictive models for Android ransomware. The results showed 98.3% accuracy even after reducing 26% of the features.

Almomani et al. [94] proposed a detection model for the Android platform based on machine learning. Ransomware samples were decompiled and a collection of features were extracted. A variety of oversampling algorithms were applied to the imbalanced dataset.

c: OTHER PLATFORMS

A novel ransomware detection model was presented by Al-Hawareh et al. [40] for IIoT systems. The model utilizes a deep learning approach along with Asynchronous Peer-to-Peer Federated Learning (AP2PFL) to provide high efficiency in detection. Chadha and Kumar [43] implemented a model that focused on discovering domains used by ransomware. Specifically, their system predicted the domains by using a real data set containing 131 malicious domains distributed into four groups. Cusack et al. [46] developed a new tool, called, programmable forwarding engines (PFEs) that can collect per-packets and monitor network data at high rates. The latter is based on using random forests and binary classifiers. Their results show that the flow-based fingerprinting method used can be accurate enough to detect ransomware before encryption. Mehnaz et al. [49] proposed a real-time ransomware detection mechanism based on deploying decoy techniques, where the running processes and the file system are monitored for malicious activities, and then benign file changes are identified and stopped from being flagged through the learning of users' encryption behavior. Azmoodeh et al. [51] proposed a unique approach of using power consumption levels as a feature to detect malicious software for Android phones. By monitoring the patterns of energy consumption, their system was able to detect crypto-ransomware families. Naik et al. [54] designed a new detection approach based on fuzzy hashing. The latter is used to cluster the so-called WannaCry or WannaCrypto ransomware using three fuzzy hashing methods, known as SSDEEP, SDHASH, and mvHASH-B. To detect ransomware, Lee et al. [55] used entropy techniques to measure the homogeneity of encrypted files. The authors leveraged machine learning and entropy analysis to classify infected files. Even if the user's system is infected with ransomware, the proposed technique can restore the original files from backups by inferring ransomware-infected files that have been synced to the backup. The analysis results confirmed that the proposed method provides a high detection rate with low error rates compared to existing detection methods.

Zhang et al. [56] proposed a ransomware classification model. Initially, N-gram sequences are created using ransomware sample opcode sequences. After that, each N-frequency-inverse gram's document is determined in order to identify feature N-grams that demonstrate more accurate family classification. Last but not least, the authors used five machine-learning techniques to classify ransomware using the results of the first step. To validate the model, six evaluation criteria are used. Extensive tests were out on actual datasets to show that the method can reach excellent accuracy. Poudyal et al. [61] have incorporated machine learning and reverse engineering to effectively detect ransomware. Their work focuses on analyzing and interpreting the malware code by performing multi-level analysis. Features such as raw binaries, function calls, and assembly codes were extracted and supervised machine-learning algorithms were employed to identify the samples. Alam et al. [63] provided an extended two-step unsupervised detection framework called RATAFIA. The model leverages a Fast Fourier Transform and a Deep Neural Network architecture to create a ransomware detection method. The suggested approach operates independently of the OS kernel. Furthermore, the authors presented a specific detection module for accurately detecting benign disk encryption processes that share traits with criminal ransomware programs but serve a distinct purpose.

Alqahtani et al. [65] proposed a framework that is intended for early detection of ransomware. The proposed model is not implemented but rather outlines the framework of the attack phase prior to encryption, taking into account the temporal relationship between IRPs and APIs, The model is presented as a high-level architecture with implementation or validation through simulation. Mary et al. [70] proposed a gradient tree boosting algorithm in order to classify a sample as malware, ransomware or benign. The proposed method performs static analysis of the samples and extracts features which are then fed to the ML algorithm. Basnet et al. [89] proposed a framework to simulate ransomware attacks in SCADA systems illustrated on electric vehicle supply equipment. The authors created a dataset of ransomware and the benign samples in the simulated environment, and then significant features are extracted to build a dataset for training and validation using the deep learning method. Almousa et al. [91] leveraged machine learning techniques on TCP malware network traffic to detect and classify ransomware families. The authors used several algorithms such as J48 decision tree and random forest to achieve their goals.

Poudyal et al. [30] developed a hybrid model powered by Artificial Intelligence. This model used several features such as function calls, assembly, and dynamic link library to overcome recent challenges in ransomware detection. A Redundancy Coefficient Gradual Upweighting (RCGU) method was put forth by Al-rimy et al. [95] to improve feature selection for crypto-ransomware detection. With this approach, the weight of the redundancy term rises proportionately to the number of characteristics that have already been chosen. A better approximation was obtained by combining the technique with additional mutual information methods. In comparison to other similar study efforts, the accuracy was also higher. Taylor et al. [37] introduced a new ransomware detection tool that uses data streams from onboard sensors to detect the start of an infection. These streams are commonly used to analyze the state of modern computing systems. There were two test systems used, one with a relatively low amount of sensor data accessible and the other with a comparatively significant amount. Berrueta et al. [79] introduced a detection approach based on file-sharing traffic analysis that can detect and stop crypto-ransomware activity. The latter uses machine learning techniques to monitor traffic between clients and file servers. Not only does it work with clear text protocols but is designed for encrypted file-sharing protocols too. Prachi and Kumar [80] presented another approach for virtual servers in organizational private clouds. The latter can extract file system, RAM, and network features after the execution of malicious and benign samples, by using machine learning and feature selection techniques. The authors showed that their approach can be an effective tool for detecting infection in organizational virtual machines. Zhang et al. [31] built a model called PreD for binary classification based on Convolutional Neural Networks. In order to improve the performance of their model, a transfer learning mechanism was employed. In their dissertation, Li and Trajkovic [83] proposed new algorithms, based on Broad Learning System, both with and without incremental learning, to classify ransomware and other types of attacks. The authors used a number of machine learning models to detect the malicious behavior of network users. He developed a so-called BGPGuard tool that integrates various stages of the anomaly detection procedure.

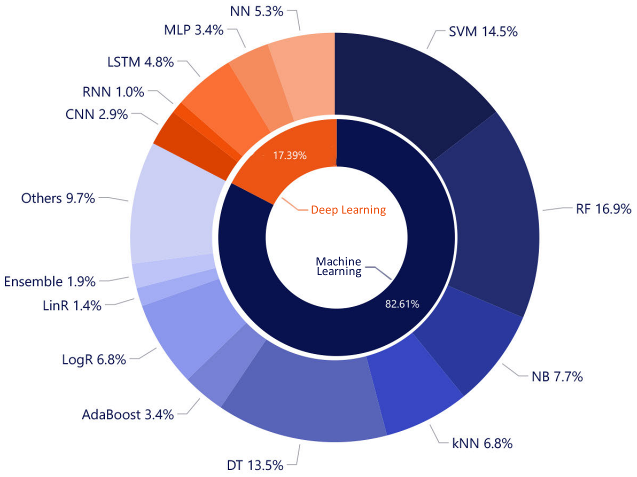

FIGURE 9. Traditional ML vs DL Approaches Used in Literature Studies.

Figure 9 illustrates the distribution of different machine learning and deep learning techniques, represented as percentages.

2) NON-MACHINE LEARNING-BASED DETECTION

Recent research in the field of cybersecurity has focused on developing effective methods for detecting ransomware attacks. One approach that has gained traction is the use of decoy-based techniques, which involve creating ''honeypot'' systems that mimic the characteristics of valuable network resources in order to lure in and detect ransomware attacks. Another approach that has been explored is the use of Software-Defined Networking (SDN) to monitor network traffic and identify malicious activity. Additionally, some researchers have proposed using rule-based methods, which rely on predefined rules or signature patterns to detect known ransomware strains. In this section, we discuss research contributions based on these approaches. Table 4 presents an overview of the non-machine learning-based research efforts aimed at detecting ransomware.

Ahmadian et al. [97] proposed a comprehensive ransomware survey. They also uncovered a method to identify resistant ransomware and stop them from encrypting victims' data based on this taxonomy and a crucial characteristic they found during the key exchange protocol phase in High Survivable Ransomware (HSR). Experimental analysis shows

TABLE 4. Summary of Non-Machine Learning-based Ransomware Detection Contributions.

Publication	Approach	Features	Performance evaluation metrics	Accuracy	Platform	Environment (Virtual/Real)
[97]	Rule-based	DNS domain requests Network traffic	Is proposed system able to detect- Yes/No	-	Windows	-
[98]	Rule-based	API calls Entropy File read/write access	Accuracy Error rate	-	Windows	Real test bed
[99]	Decoy-based	File access	Accuracy Precision Recall	-	IoT	Real custom test bed
[100]	Rule-based	RST packets, HTTP traffic	-	-	Windows	-
[101]	Rule-based	File entropy, type changes, similarity File deletion, file type funneling API calls	-	-	Windows 7	Cuckoo
[38]	Hardware-based	I/O requests	Accuracy	96.3%	Windows	Cuckoo
[93]	Decoy-based	Low level file system activity I/O-level activity User data	-	-	Windows 7	-
[102]	Decoy-based	File accesses	-	-	Windows	-
[34]	Decoy-based	File access	Accuracy Effectivity Low consumption Plainness Simplicity	100.0%	Ubuntu 16.04	-
[103]	Rule-based	TCP/IP header	-		IoT	
[104]	Rule-based	Images and texts from the XML layout files, resources, and classes.dex	Accuracy	91.0%	Android	Four Nexus 7 tablets running Android 6.0.1 Marshmallow
[105]	FSM based	Intensity, similarity of files Entropy	Accuracy	-		DiskSim augmented SSD simulator
[106]	Rule-based	IP traffic	Detection rate, FPR # files lost, Overhead,	100.0%	Windows 7	Virtual
[107]	SDN-based	HTTP messages	ROC curve	-	Windows 7	Cuckoo VMware ESXi hypervisor
[108]	Forensics- based	Assembly instructions, function calls Network signatures	-		Windows	Real test bed
[109]	Rule-based	Random and Gibberish Characters in the Domains Frequency of Different Domains Generation The Replication of the Same Domains in a Time Interval	Accuracy, FPR, FNR	100%	Windows 7	
[110]	Rule-based	UI indicators (File list, Hint text, Button)	Accuracy Effectiveness Runtime performance Analysis time Memory consumption	99.0%	Android	Nexus 5 Android 6.0
[111]	FSM-based	API calls	Accuracy, min, max time for analysis	-	Android	-
[112]	Rule-based	Entropy signature DLL monitoring, API calls Windows Registry	Context-aware trigger (CAT) thresold	-	Windows	Cuckoo
[113]	Rule-based	File similarity	-	-	-	-
[114]	FSM-based	Changes in user files Persistence of active programs Lateral movement System resources.	Accuracy TPR, FNR	99.5%	Windows 7	Real custom - C#
[115]	Decoy-based	File access (r/w/d/e)	Accuracy Precision Recall	96.2%	IoT Android	Real Pixel mobile device with Android 7.1
[116]	Decoy-based	-	# samples detected		Windows	Malboxes Vagrant

that the framework is capable of spotting variations of current ransomware. Scaife et al. [101] presented CryptoDrop, which is an early detection system that can notify against ransomware activities. The system leveraged behavior indicators to stop suspicious processes with low false positives. Based on dynamic analysis, Kharaz et al. [38] presented UNVEIL, a system dedicated to ransomware detection. The detection is based on tampering files (e.g., desktop and/or user files). The system generates an artificial user environment and identifies ransomware modifications on files. Furthermore, the system tracks system changes and behavior for changes. The detection is able to identify unknown (zero-day) evasive ransomware that has not been previously reported. ShieldFS was proposed by Continella et al. [93] as a defense mechanism against ransomware threats. By keeping an eye on file system activity and creating trustworthy profiles, the system automatically defends against attacks. When a profile behaves abnormally, its activities are flagged as malicious, and any negative consequences it may have on the file system are transparently undone. The system was developed using an analysis of billions of I/O file system requests made by good programs and gathered from good computers over the course of a month. ShieldFS was examined in actual operations, tested against different malware (ransomware) families, and shown to be effective in spotting threats and efficiently recovering files.

Moore [102] proposed a honeypot-based ransomware detection model for Windows networks. By placing decoy files across the network, the model is able to detect activity on these files and send email alerts to users. Hernádez et al. [34] proposed a decoy-based detection and prevention system. By deploying a set of honey files across the user environment, the system blocks the process that tries to read these decoy files. In addition, the system automatically begins corrective steps to stop the infection. Zahra and Shah [103] employ a black listing of Command and Control servers by monitoring the network traffic generated by ransomware. The proposed model is able to detect Cryptowall ransomware attacks in IoT environments. A model for ransomware security for Android devices was proposed by Alzahrani et al. [104]. Their system, RanDroid, analyzes the image textural strings and other information from the app and compares it to a set of predefined information stored in a database. This database is built using information extracted from known variants of ransomware. A set threshold is used to decide whether an app is malicious or not. Min et al. [105] proposed Amoeba, an SSD system that supports automated backup, to fight against ransomware attacks. The system is equipped with a hardware accelerator that can detect the infection of pages through a prompt backup control mechanism. The purpose is to minimize space overhead for original data backup. Microsoft SSD Simulation has been used for evaluation and real block-level traces are used from dynamic analysis. The system outperformed FlashGuard, another system that supports data backup within the device.

Morato et al. [106] employ a network prober to passively monitor the traffic generated by 19 ransomware families. The model focuses on early detection and less than 10 files are lost before the model detects the ransomware activities. The authors were able to recover these lost files from the data monitored by the prober. Cabaj et al. [107] presented a detection method based on Software-Defined Networking (SDN) for Windows machines. The method monitors the network traffic between two crypto-ransomware variants, CryptoWall and Locky, and extracts relevant HTTP message sequences as the deciding features. Subedi et al. [108] employed both static and dynamic analysis of the malware executable to derive static and run-time behavioral features. The model called CRSTATIC uses reverse engineering to extract signatures of the binary. The work proposed by Salehi et al. [109] focuses on detecting bots and DGA-based ransomware. The authors have utilized behavioral features and calculated the metrics namely, the generation frequency of different domains and the repetition of a particular domain in a given time period. Chen et al. [110] focused on providing ransomware security for Android devices. By collecting a large dataset that covers the majority of ransomware families for Android, their proposed system, RansomProber, was able to detect malicious apps with allowable run-time performance and high accuracy. Junaid et al. [111] proposed StateDroid, a two-layer finite state machine model (FSM) to represent the sneaky Android app attacks as state transitions. Jung and Won [98] monitor the read/write activity of ransomware samples and backup files that have a large read/write operation. The context-aware detection model uses entropy information in order to identify abnormal file activity. Almashhandani et al. [100] considered the dynamic analysis of Locky ransomware. An intrusion detection system was developed based on the extracted network traffic features in order to detect the crypto-ransomware family. The authors demonstrate how network traffic can be used for early ransomware detection.

Singh et al. [112] developed a detection mechanism based on a set of features such as dynamic link libraries, API calls, registry, and entropy of files. The authors further presented how this context-aware detection method can be integrated into digital forensics for ransomware mitigation and prevention. Almashhadani et al. [100] proposed a network-based intrusion detection system that can effectively track ransomware network activities with high accuracy and a low false positive rate. The authors used, as a case study, the so-called Locky, one of the most dangerous families of crypto-ransomware. Naik et al. [113] analyzed malware in its early stages and compared it to known malware using fuzzy hashing, YARA rules, and import hashing. The authors showed that the YARA rules with fuzzy hashing can yield improved assessment results, irrespective of the malware type. By examining resource utilization, persistence, lateral movement, and user files, the finite state machine-based model implemented by Ramesh and Menen [114] is able to detect different variants of ransomware. Chakkaravarthy et al. [99] developed an Intrusion Detection Honeypot (IDH) for ransomware detection that deploys the so-called Social Leopard Algorithm to give early warnings in the presence of suspicious files. Their experiments showed that IDH outperforms previously known ransomware detection models in terms of detection time and accuracy. Wang et al. [115] examined the traits of cryptographic ransomware. They suggested a decoy-based file protection technique against ransomware. They created and deployed KRProtector to identify ransomware and protect files based on decoys in order to meet the requirement for file protection on devices without roots. Gómez-Hernández et al. [116] introduced and extended R-Locker. The tool is built on a honey file-based strategy, in which trap files are dispersed around the target file system to help detect and prevent ransomware promptly. The tool was expanded by the authors in different ways. R-Locker is first ported to Windows platforms, where there are unique peculiarities in FIFO handling. Second, to maximize protection, the honey files' overall management surrounding the target disk has been upgraded. Finally, using the dynamic white and black lists dataset, suspected malware is near-real-time blocked. The new version of R-Locker exhibited remarkable effectiveness and efficiency in combating ransomware.

Summary of findings in Section IV-A:

The detection studies can be broadly classified as those that focused on detection using ML approaches and those that did not employ ML algorithms. In this section, we conclude our findings of the research work on ransomware detection. We can draw the following insights:

- Machine learning-based approaches were a clear choice among researchers with over 70% using either traditional ML classifiers or DL techniques to detect ransomware. This is due to ML algorithms' high accuracy rate, efficiency as well as speed.

TABLE 5. Summary of Ransomware Classification Contributions.

Publication	Approach	Features	ML technique		Accuracy	Dataset		
			ML classifier	DL technique		Source	# Ransomware families	# Samples
[13]	ML-based	API calls	-	DNN	95.96%	VT VS Malwarebytes Offensive-computing	14	483 R 754 B
[117]	ML-based	API calls	-	LSTM	96.7%	-	15	157 R
[118]	ML-based	File Access Created Processes Required permissions Injected Process Shell Commands Searched Windows Opened Service Managers Created Mutexes Opened Mutexes Numeric Open Services Runtime DLL	DT J48 kNN NB	-	78%	-	10	150 R
[119]	ML-based	Tweets	-	DNN	78.9%	-	25	-
[120]	ML-based	System calls	-	LSTM CNN	-	-	3	660 R 219 B
[121]	ML-based	IP File length URL	RF SVM GTB	-	98.5%	-	-	40K R 25K B

- Among the traditional ML classifiers that were used, SVM and RF were the most preferred classifiers by researchers. SVM works by finding the best boundary, or hyperplane, to separate the data into different classes. Random Forest, on the other hand, is an ensemble learning method. SVM is particularly effective for ransomware analysis due to its ability to handle high-dimensional and non-linearly separable feature sets, while RF is robust to over-fitting, and is less likely to be affected by outliers or noise in the data.
- Majority of the ransomware detection works were for the Windows platform followed by Android reflecting the fact that Windows OS is the most popular target of ransomware attacks.

B. CLASSIFICATION

Ransomware classification involves identifying and categorizing malware samples based on their characteristics and behavior. This can be done by analyzing properties such as code patterns, network communications, and system call sequences, and comparing them to known malware families to determine similarities and assign a classification. The use of signatures, whether they are static or dynamic, can aid in this process to help detect and prevent the spread of malware. Table 5 provides a summary of the research contributions focused on classifying and categorizing the various strains of ransomware.

Maniath et al. [117] introduced an automated ransomware detection technique, based on dynamic analysis. The latter extracts from the logs Application Programming Interface (API) call to detect ransomware. The authors showed that their approach can improve the automatic analysis of ample malware samples. Wani and Revathi [122] employed machine learning to classify ransomware samples into different variants. The authors analyzed the behavioral characteristics of 150 ransomware samples in order to provide the classification. Daku et al. [118] employed machine learning to classify ransomware samples into different variants. The authors analyzed the behavioral characteristics of 150 ransomware samples in order to provide the classification. Vinaykumar et al. [119] employed a deep learning approach to classify ransomware-related tweets to their respective families. Posts from social media platforms are constantly monitored and the incident response team is alerted about the spread of ransomware. Homayoun et al. [120] proposed a new method for analyzing ransomware, which they referred to as the Deep Ransomware Threat Hunting and Intelligence System (DRTHIS). The latter is based on using convolutional neural networks, long short-term memory, and two deep learning approaches. The authors showed that DRTHIS can achieve an F-measure of 99.6% with a positive rate of 97.2%. They also revealed how DRTHIS can detect unseen ransomware from a number of new ransomware families in a reasonable amount of time. By employing semi-supervised deep learning approaches, Sharmeen et al. [13] identified deviating patterns in new ransomware variants. Behavioral attributes of the samples are extracted and the proposed deep learning approach is integrated with supervised learning to make the system robust. The model is also capable of detecting zero-day variants. Usharani et al. [121] developed a ransomware classification method using a machine learning algorithm achieving 98.45% accuracy and a 0.01 false rate. The data, of crypto-ransomware type, was collected and analyzed dynamically. The authors performed comparisons to Linear Regression, Adaboost, and Naive Bayes.

Summary of findings in Section IV-B:

This section discusses the classification process of categorizing ransomware based on its characteristics and behavior.

By exploring this section, we can draw the following key conclusions:

- All the studies that dealt with the classification of ransomware samples used machine learning-based approaches. This is due the Machine learning's ability to automatically and efficiently analyze large amounts of data and provides a scalable solution to the problem of increasing malware variants and infection rates.
- There were two main categories of classification- Works that classified a sample as either ransomware or benign and those that classified a sample into different ransomware variants and/or families.
- Finally, all the classification studies have also aimed to detect ransomware. Ransomware detection and ransomware classification are two distinct but related processes in that detection provide the input for classification. Once ransomware is detected, it can be analyzed and classified to gain a deeper understanding of its behavior and impact.

C. PREVENTION

Ransomware prevention involves a variety of techniques and measures to protect networks, systems, and data from ransomware attacks. Table 4 offers a summary of the research contributions focused on preventing ransomware attacks presented in various publications and studies. In this section, we delve into various prevention techniques proposed by literature studies. Table 6 presents a summary of the publications that focus on ransomware prevention.

Song et al. [125] proposed a prevention method for Android devices. Using statistical methods, the system monitors features such as I/O rates, memory consumption, and processor usage in order to detect any process with unusual behaviors. If such processes are detected, the system removes the associated program from the machine. Lee et al. [127] designed a real-time prevention framework based on detection in cloud analysis and abnormal behavior analysis. The latter gathers information from devices as well as logs and analyzes them using a cloud system, hence minimizing early intrusions. Kharraz and Kirda [124] proposed a dynamic analysis model for ransomware detection. The method is based on observing and interacting with a user's files or desktop. The system detects when the ransomware first interacts with user data and tracks changes to the system's desktop. These patterns are used to identify ransomware-like activities. The authors applied the system to the new ransomware family. Kim et al. [130] proposed an outline of a ransomware detection model based on the random number generated by the user's OS. The model is designed to work for ransomware that uses the CryptGenRandom() to generate the random number and the encryption key is recovered after an infection occurs. Ami et al. [128] proposed a system, Antibiotics, that provides ransomware prevention by imposing a file-access control policy. This system prevents malicious software from modifying and deleting user files by providing biometric authentication and schemes such as CAPTCHA to determine if a user is human or not. Shaukat and Ribeiro [123] designed a layered defense system called RansomWall for crypto-ransomware families. It employs hybrid analysis to generate a set of features that is characteristic/typical of ransomware actions. If a process is tagged as suspicious by the system, all files modified by this process are backed up. Lee et al. [19] focused on the backup of the encryption keys in a safe repository. Such a technique enables the systems or files infected by ransomware to be easily recovered thereby ensuring protection against malicious attacks. Lee et al. [126] applied the concept of moving target defense (MTD) by which the user files can be protected even if attackers constantly change their encryption tactics. The authors have demonstrated that by randomly and endlessly changing the extensions of files, the proposed system can successfully provide ransomware protection.

Because malware can easily identify virtual and analysis environments, Zhang et al. [39] used a deception technique to enhance the analysis of malware by building a more secure (less weak against environment detection) system. In order to achieve this goal, the authors proposed Scarecrow, a lightweight deception engine that successfully deactivates samples of evasive malware. AlSabeh et al. [129] analyzed certain actions that ransomware takes in order to detect its execution surroundings. The proposed approach is able to detect if a process is trying to detect its environment by intercepting the Windows API calls. Finally, the approach aborts any such process and prevents an attack. Wani and Revathi [122] focused on ransomware-threatening IoT. In fact, they presented a detection model for IoT ransomware attacks. The proposed solution proposes utilizes Software Defined Network (SDN) gateway to closely monitor incoming traffic within IoT systems. Furthermore, it detects and mitigates IoT ransomware with policies in the SDN controller.

Summary of findings in Section IV-C:

In this section we discussed the contributions that included enforcing file access policy, backing up encryption keys, and the application of Moving Target Defense (MTD) among others. Some of the insights that can be drawn from the research on ransomware prevention include:

- Majority of the contributions focused on preventive ransomware solutions for Windows operating systems. Historically, Windows has been the most targeted operating system due to its widespread usage and popularity. Windows is used on over a billion personal computers and is the dominant operating system in businesses and organizations, making it a prime target for attackers looking to exploit a large number of systems. Additionally, Windows is known to have had vulnerabilities in the past, and it can be more difficult to secure due to its complexity and the sheer volume of software that runs on it.
- More focus was placed on performing dynamic analysis as this type of analysis can determine the impact of malware and inform more effective prevention

TABLE 6. Summary of Ransomware Prevention Contributions.

Publication	Type of Analysis	Features	Performance evaluation metrics	Accuracy	Platform
[19]	Dynamic	Dll calls	-	-	Windows
[122]	Static	Network traffic: Constrained Application Protocol (CoAP) headers TCP/IP headers	TPR, FPR, Accuracy P, F1, MCC	97.9%	IoT
[123]	Dynamic	APIs system calls	Accuracy	98.3%	Windows
[124]	Dynamic	I/O access	TPR, FPR	-	Windows
[39]	Dynamic	Windows kernel activities: file system I/O, registry process/thread creation and termination network activity, DLL loading/unloading	% of deactivated evasive malware	89.6%	Windows 7
[125]	Dynamic	Processor usage Memory usage I/O rates	Protection without ransomware information Operates without updating Operates without downloading Operates without executing application	-	Android
[126]	Dynamic	-	# Ransomware samples that were prevented from encrypting files	98.6%	Windows 7
[127]	Hybrid	Network traffic, file operations, file size File signature, generation time Activities carried out on servers, server's activities, Monitoring malicious code that access server	Accuracy Reliability Availability	-	Cloud
[128]	Dynamic	I/O requests System calls	-	-	Windows
[129]	Dynamic	API calls	Accuracy	91.0%	Windows 10
[130]	Dynamic	API calls	-	-	Windows

strategies, such as validating security solutions, identifying potential threats, and improving incident response plans.

D. MITIGATION

In recent years, researchers have been working to develop effective methods for mitigating the impact of ransomware attacks. One approach that has been proposed is the use of key-escrow, which involves storing a copy of the encryption keys used by ransomware in a secure location. This allows organizations to recover their data in the event of an attack, without having to pay the ransom. Another popular approach is the use of Software-Defined Networking (SDN) to detect and block malicious traffic at the network level. Additionally, some researchers have proposed using forensic-based methods, which involve collecting and analyzing data from infected systems to identify the cause of the attack and to track the attackers. Another approach is the use of sensor-based methods, which involves deploying various types of sensors to detect the presence of ransomware on a network and to generate alerts in real-time. Table 7 summarizes the research efforts aimed at mitigating the impact of ransomware, as presented in various publications.

Cabaj and Mazurczyk [134] presented two mitigation models. The proposed system first monitors the network traffic for any suspicious activities and then provides real-time mitigation by blocking off the infected hosts using control rules. Kolodenker et al. [132] proposed Paybreak, which fights ransomware by keeping by holding encryption keys in escrow and allowing victims to restore encrypted files without paying the ransom. Monge et al. [137] stressed on monitoring the environment without the involvement of a human operator. Their model is based on a self-organizing network framework and involves mitigation against crypto-ransomware families that contact suspicious C&C servers for encryption keys. Aidan et al. [133] introduced a new and enhanced version of the Petya ransomware along with two mitigation strategies to defend against it. The multi-layered security approach is expected to reduce the rate of successful ransomware attacks as cybercriminals continue to improve their attack methods.

Maimó et al. [138] highlighted the importance of ransomware security in Integrated Clinical Environments (ICE). Their proposed solution employs an SDN framework along with Network Function Virtualization (NFV) to mitigate the propagation by replacing and isolating infected systems. Another work that provided ransomware mitigation using software-defined networking was the one proposed by Akbanov et al. [136]. The model is capable of blocking infected hosts in real-time by monitoring network traffic for suspicious file activities. The authors considered the WannaCry sample for the evaluation of the model. Considerable static and dynamic analysis of ExPetr ransomware was conducted by Rouka et al. [135]. Their SDN-based system focused on three areas for mitigation namely, port blocking, HTTP packet inspection, and examination of SMB messages. Davies et al. [139] utilized live forensics tools to examine the memory captured from a ransomware-infected system. The goal was to decrypt the files encrypted by NotPetya, Bad Rabbit, and Phobos ransomware using the keys found in the examined memory. A data-centric mitigation and detection technique was proposed by Faghihi and Zulkernine [131] to defend against crypto-ransomware attacks for smartphones.

TABLE 7. Summary of Ransomware Mitigation Contributions.

Publication	Approach	Type of Analysis	Features	Platform	Experimental Environment (Virtual/Real)
[131]	Signature-based Anomaly-based	Hybrid	API calls Entropy File structure	Android	Android 8 emulator
[132]	Key escrow	Dynamic	-	Windows 7	Cuckoo
[133]	Rule-based	Dynamic	-	-	-
[134]	SDN-based	Dynamic	DNS messages		Xen
[135]	SDN-based	Dynamic	SMP messages HTTP messages	-	Virtualbox
[136]	SDN-based	Hybrid	Hashes, dlls DNS,TCP traffic	-	-
[137]	Sensor-based	Dynamic	System calls Handshakes for exchanging keys	-	-
[138]	ML-based SDN-based	Dynamic	Traffic capture - Start time, protocol (UDP/TCP/ARP) Flow duration (s) Source IP, source port, direction Destination IP, destination port, state Total packets, total bytes Source packets, source bytes Source load (bits/s), total load (bits/s) Source inter-packet arrival time (msec) Destination inter-packet arrival time (msec)	Integrated Clinical Environments (ICE)	OpenICE
[139]	Forensics-based	Dynamic	Memory dumps- keys	-	Real custom
[140]	SDN-based	Hybrid	Static IOCs Network traffic Network protocols	-	Virtualbox

The model, called Ransomcare, performs real-time security through hybrid analysis of the samples and recovery of lost files through backups. Umar et al. [140] focused on mitigating ransomware attacks on cloud networks. This approach statically and dynamically analyzes the Sodinokibi ransomware and suppresses TCP access on the infected networks.

Summary of findings in Section IV-D:

In the mitigation section, we covered key contributions including the use of key escrow techniques, the employment of software-defined networking to block malicious traffic, and performing forensic analysis of infected systems. By reviewing Section IV-D, we can summarize the following conclusions:

- Majority of the mitigation papers proposed approaches using Software-defined Networks (SDN) as SDN improves malware mitigation through centralized network management and increased visibility into network activity.
- All the studies performed a dynamic analysis of the ransomware samples to extract behavioral features.
- Network traffic characteristics were gathered and relevant features were extracted by the majority of the papers as it provides valuable information about the presence and spread of malware, helps organizations determine the origin and scope of an outbreak, and enable security teams to prevent malware from entering the network.

E. PREDICTION

Ransomware prediction refers to the ability to detect the potential occurrence of a ransomware attack before it happens. This can allow organizations to take proactive measures to protect themselves and minimize the potential impact of the attack. Machine learning-based methods have been proposed for predicting ransomware attacks. These methods involve using algorithms such as artificial neural networks, decision trees, and support vector machines to analyze data from various sources and detect patterns that may indicate an impending ransomware attack. By training these algorithms on historical data, researchers aim to develop models that can accurately identify the signs of a ransomware attack in real-time. Table 8 provides a summary of the research contributions directed towards predicting ransomware attacks presented in various publications and studies.

Quinkert et al. [146] proposed a model to predict whether a newly registered domain is going to be used in a ransomware attack. The model utilized two supervised machine learning classifiers that use different feature sets from the same training dataset. The time series forecasting method is employed to predict future ransomware activities. Rhode et al. [143] examined the feasibility of determining if an executable is harmful by analyzing a small segment of its behavioral data. The authors used a combination of recurrent neural networks that allowed for accurate prediction of maliciousness within the initial 5 seconds of execution. Adamu and Awan [147] used machine learning techniques to detect ransomware. They used selected attributes in their dataset to predict attacks. Support Vector Machine has been found to be more efficient when compared with other machine learning classifiers. Chang et al. [145] implemented a prediction system that adopts a k-nearest neighbor algorithm to detect and predict ransomware network traffic. By employing a static analysis approach, the system monitors unknown IP traffic that is an indication of malicious activity.

TABLE 8. Summary of Ransomware Prediction Contributions.

Publication	Approach	Type of Analysis	Features	ML/DL technique
[141]	ML-based	Dynamic	IP ports	SVM
[142]	ML-based	No analysis	Bitcoin addresses	XGBT RF
[143]	ML-based	Dynamic	File machine activity	Ensemble RNN
[144]	ML-based	Static	Bitcoin transaction features: Address, day, year, Weight, length, count, looped, Neighbors, label, income	LogR, DT, RF Boosting
[145]	ML-based	Static	IP addresses	kNN
[146]	Probability-based	Static	length of domain, domain name Domain registrant, admin Time of registration	Yes Did not mention which
[147]	ML-based Decoy-based	Dynamic	File accesses	SVM, RF, DT BN, LogR, ANN

Akcora et al. [142] presented a new approach that utilizes the latest advancements in Topological Data Analysis (TDA) to effectively and easily predict potential new ransomware transactions within a specific family. The proposed method only requires a minimal amount of historical transaction data. Mathane and Lakshmi [141] proposed a solution for predicting ransomware attacks in industrial IoT systems. The model which is based on context awareness employs SVM to make early predictions of an attack. Xu [144] utilize a vast amount of features from Bitcoin transactions. The authors conducted descriptive statistical analysis and utilized machine learning models to construct a prediction model for identifying and classifying ransomware families, with the goal of preventing financial loss from ransomware attacks.

Summary of findings in Section IV-E: In the last section of ransomware prediction, we have discussed crucial contributions such as the time series forecasting method, topological data analysis, and statistical analysis of bitcoin transactions. Based on the information discussed in Section IV-E, we can summarize the following key points:

- Most researchers preferred machine learning-based approaches for predicting ransomware, with 86% of the studies using machine learning algorithms.
- An equal number of studies utilized both static and dynamic analysis techniques.
- Traditional machine learning classifiers were the preferred choice for ransomware prediction systems compared to using neural networks. This could be due to several reasons. Traditional machine learning algorithms are computationally more efficient than neural networks, which is important for resource-limited scenarios or real-time predictions. Neural networks often require large amounts of data to train effectively, while traditional machine learning algorithms can work well with smaller datasets.

V. DISCUSSION

According to our readings, we have chosen the below topics for discussion. The first part includes a discussion on the trend of malware and the lack of cutting-edge research. The second elaborates on adversarial machine learning attacks. The aim of this section is to highlight some findings and interpret them. This could be a great guideline for researchers and an opportunity for ransomware future works.

- **Malware Trends and the lack of cutting-edge research:** In our survey, we have identified several research trends. In fact, we have noticed that the majority of research contributions are heading toward the detection of ransomware using machine-learning models (as per Section IV-A2). In addition, the majority of these publications, which fall under the detection of ransomware using machine learning, are more theoretical than practical. For instance, many authors proposed models and approaches without inferring the detection rate, which is considered critical for ransomware prevention. Recall, no matter which detection technique you leverage, it is useless to detect if the prevention, which is based on this detection, occurs after the infection. Moreover, other researchers have not implemented any approaches and just discussed proposals. Having said that, we have identified a lack of cutting-edge research contributions, which possess a high impact on solving the ransomware problems. For instance, rather than solely detecting malware, we foresee that building real-time protection capabilities such as in [101] and identifying and preventing zero-day ransomware could be a core solution to ransomware defense and promising research directions. Another lack of research found in our survey is the low number of contributions that leveraged reverse engineering techniques on the top ransomware in the past couple of years. For instance, the ransomware Conti, Hive, Revil, Lockbit 3.0, and Egregor are among the top ransomware that, in regard to cost impact since 2020, have not been thoroughly investigated. In our opinion, the complexity of these malware is one reason why they are still active for several years and have not been scrutinized thoroughly. In a nutshell, the majority of the identified techniques fall under the machine learning detection of ransomware and have a weak impact on the effectiveness of their approach in preventing ransomware in general and zero-day (unknown) threats in particular. Although prevention of zero-day attacks will remain one of the most challenging tasks for security professionals and the widespread use of smart and Internet of Things (IoT) devices has made it increasingly challenging for security experts to effectively guard against the threats posed by ransomware, there are still critical research directions for improving ransomware defense solutions.

- **Adversarial machine learning and future research directions:** While Adversarial Machine Learning (AML) refers to the attacks that manipulate the decision-making process of machine learning models by feeding them misleading input, it also refers to a sub-field of AI that focuses on the design of machine learning models that can defend against these malicious attacks and the ways in which these models can be made robust against such attacks. There are two primary types of Adversarial ML attacks namely, evasion and poisoning attacks. Evasion attacks modify the input data, for example, by tweaking the code structure, or the file attributes, and are able to make the model incorrectly classify the input data, thereby evading detection. These types of attacks target a model that has already been trained. Poisoning attacks, unlike evasion attacks, manipulate the training dataset used to train the machine learning model.
  By designing robust models against adversarial attacks, we can ensure that machine-learning systems are trustworthy and reliable, and can be safely used in critical applications. We present some of the contributions that defend against adversarial attacks on machine learning used for classifying malware and publications that propose adversarial models to attack machine learning classifiers. For instance, Chen et al. [148] present an evasion attack model called EvnAttack, for a learning-based classifier that uses Windows API calls features obtained from PE files. To tackle the issue of evasion attacks, the authors propose a secure-learning paradigm named SecDefender. This approach includes classifier retraining and a security regularization term to improve the system's security against feature manipulation by attackers. Furthermore, Chen et al. [110] focus on the issue of malware affecting the accuracy of machine learning classifiers. To address this problem, the authors propose a two-phase approach called KUAFUDET for detecting mobile malware. This approach involves an offline training phase to select features and an online detection phase to utilize the trained classifier. The approach also includes a self-adaptive learning scheme to continually improve accuracy by filtering false negatives and feeding them back into the training phase. Moreover, Kolosnjaji et al. [149] explored the susceptibility of deep learning approaches for detecting malware. A gradient-based attack method is introduced, which can fool a deep network-based malware detection model by making small modifications to specific bytes at the end of each malware sample while still maintaining its harmful functionality. Last but not least, Chen et al. [150] proposed an approach for creating adversarial examples of Android malware to bypass current detection models. The authors developed a tool that can automatically generate these adversarial examples, which can even fool machine learning-based detectors that use semantic features. The method was tested on two leading Android malware detection systems, Drebin and MaMaDroid, and was shown to be effective. Based on our research, we have noticed that these adversarial machine-learning approaches are the only relevant contributions in the field of malware. However, no research contributions investigated this adversarial approach to ransomware. As such, we believe that this niche of research is still very young and we are expecting more contributions in this field in the near future. Knowing that ransomware remains a type of malware, the adoption of the aforementioned adversarial machine-learning approaches could be also successful against ransomware defense.
- **Concept drift in Machine Learning:** Machine learning is currently the most investigated and utilized research technique for various topics in general and malware detection in particular. However, an issue that is relevant to machine learning models is concept drift. It is a phenomenon in machine learning that occurs when the statistical characteristics of the target variable, which the model tries to predict, undergo changes over time. As a result, the meaning of the past input data, on which the model was initially trained, has significantly evolved over time. Furthermore, it may no longer be relevant to the new or current data thereby causing the model to make inaccurate or poor predictions (e.g., unable to classify or detect ransomware in a detection model). This can have a particularly detrimental impact on critical applications and infrastructures (e.g., finance and healthcare, transportation, telecommunication, etc.). Although there has been a rise in the focus on handling concept drift, it still lacks in certain aspects. The current state of research lacks a thorough examination of frameworks, benchmarks, and real-world data streams in terms of severity of drift, occurrence time, and regions affected by the drift. Regarding research contributions, some works have explored topics such as tracking drift in various malware families using the feature type that drifts the least, classifying malware in the presence of drift by leveraging conformal evaluation to develop rejection strategies and improve security detection pipelines, and designing a feature selection architecture for ransomware detection to identify a set of features that can improve the durability and effectiveness of the machine learning model.

While some research has explored concept drift, only a handful have examined its implications for detecting and mitigating ransomware. Given that ransomware's primary defense is protection rather than detection and mitigation, we believe that this area requires greater attention in the coming years. We encourage researchers to explore the potential of dynamic machine learning algorithms with drifting capabilities for real-time protection against ransomware.

VI. CONCLUSION

Technology has impacted every part of our life in the modern world. Unfortunately, adversaries are misusing technology to their own ends. Internet services have thus developed into an accessible tool for attackers to generate destructive actions like infecting victims' computers, seizing control, depleting resources, and stealing data. Today, one of the top threats to security in large-scale organizations and governments is ransomware, a special type of malware family, designed to encrypt and lock victims' machines for ransom. In this comprehensive survey paper, we provide an overview of ransomware, including its history and evolution, taxonomy, and state-of-the-art research. By tracing the origins of ransomware and its evolution over time, this paper has highlighted key milestones and major trends in the field. The proposed taxonomy of ransomware has categorized various types of ransomware based on their defense mechanisms, characteristics, and behavior. Additionally, this paper has reviewed a total of 125 research contributions which include detection, prevention, mitigation, and prediction techniques. We have found that the research papers that cover detection dominate the publications in this domain. In particular, leveraging machine learning for the detection of ransomware surpassed all other techniques. We have revealed several classifications, such as prediction, that were unclassified in similar surveys. Furthermore, we uncover future research directions such as adversarial machine learning which has never been used in ransomware research. Finally, we trust that this survey will be a useful resource and guideline for researchers and practitioners working in the field of ransomware and will inspire future research in this area. Despite the significant progress made in the field, much more remains to be explored in the realm of ransomware research.

REFERENCES

[1] Q. Chen and R. A. Bridges, "Automated behavioral analysis of malware: A case study of WannaCry ransomware," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl.*, Dec. 2017, pp. 454–460.

[2] H. S. Lallie, L. A. Shepherd, J. R. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bellekens, "Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic," *Comput. Secur.*, vol. 105, Jun. 2021, Art. no. 102248.

[3] M. Paquet-Clouston, B. Haslhofer, and B. Dupont, "Ransomware payments in the Bitcoin ecosystem," *J. Cybersecurity*, vol. 5, no. 1, 2019, Art. no. tyz003.

[4] D. Su, J. Liu, X. Wang, and W. Wang, "Detecting Android locker-ransomware on Chinese social networks," *IEEE Access*, vol. 7, pp. 20381–20393, 2019.

[5] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the gordian knot: A look under the hood of ransomware attacks," in *Detection of Intrusions and Malware, and Vulnerability Assessment* (Lecture Notes in Computer Science) vol. 9148. Cham, Switzerland: Springer, 2015.

[6] A. Braue, "Global ransomware damage costs predicted to exceed $265 billion by 2031," Cybercrime Mag., Melbourne, VIC, Australia, Tech. Rep., 2022.

[7] M. Humayun, N. Jhanjhi, A. Alsayat, and V. Ponnusamy, "Internet of Things and ransomware: Evolution, mitigation and prevention," *Egyptian Informat. J.*, vol. 22, no. 1, pp. 105–117, 2021.

[8] I. Nadir and T. Bakhshi, "Contemporary cybercrime: A taxonomy of ransomware threats & mitigation techniques," in *Proc. Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Mar. 2018, pp. 1–7.

[9] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Comput. Secur.*, vol. 111, Dec. 2021, Art. no. 102490.

[10] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Prevention of crypto-ransomware using a pre-encryption detection algorithm," *Computers*, vol. 8, no. 4, p. 79, Nov. 2019.

[11] R. Moussaileb, R. E. Navas, and N. Cuppens, "Watch out! Doxware on the way...," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102668.

[12] P. H. Meland, Y. F. F. Bayoumy, and G. Sindre, "The ransomware-as-a-service economy within the darknet," *Comput. Secur.*, vol. 92, May 2020, Art. no. 101762.

[13] S. Sharmeen, Y. A. Ahmed, S. Huda, B. S. Kocer, and M. M. Hassan, "Avoiding future digital extortion through robust protection against ransomware threats using deep learning based adaptive approaches," *IEEE Access*, vol. 8, pp. 24522–24534, 2020.

[14] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A survey on ransomware: Evolution, taxonomy, and defense solutions," *ACM Comput. Surv.*, vol. 54, no. 11, pp. 1–37, Jan. 2022.

[15] F. Khan, C. Ncube, L. K. Ramasamy, S. Kadry, and Y. Nam, "A digital DNA sequencing engine for ransomware detection using machine learning," *IEEE Access*, vol. 8, pp. 119710–119719, 2020.

[16] B. Zhang, W. Xiao, X. Xiao, A. K. Sangaiah, W. Zhang, and J. Zhang, "Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes," *Future Gener. Comput. Syst.*, vol. 110, pp. 708–720, Sep. 2020.

[17] A. Zimba, Z. Wang, and H. Chen, "Reasoning crypto ransomware infection vectors with Bayesian networks," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 149–151.

[18] I. Bello, H. Chiroma, U. A. Abdullahi, A. Y. Gital, F. Jauro, A. Khan, J. O. Okesola, and S. M. Abdulhamid, "Detecting ransomware attacks using intelligent algorithms: Recent development and next direction from deep learning and big data perspectives," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 9, pp. 8699–8717, Sep. 2021.

[19] K. Lee, K. Yim, and J. T. Seo, "Ransomware prevention technique using key backup," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 3, p. e4337, 2018.

[20] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, *Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection*. Cham, Switzerland: Springer, 2018, pp. 93–106.

[21] *Darkside Ransomware: Best Practices for Preventing Bus. Disruption From Ransomware Attacks*, CISA and FBI, Dept. Homeland Secur., Cybersecur. Infrastruct. Secur. Agency, Washington, DC, USA, 2021.

[22] D. W. Fernando, N. Komninos, and T. Chen, "A study on the evolution of ransomware detection using machine learning and deep learning techniques," *IoT*, vol. 1, no. 2, pp. 551–604, Dec. 2020.

[23] E. Berrueta, D. Morato, E. Magana, and M. Izal, "A survey on detection techniques for cryptographic ransomware," *IEEE Access*, vol. 7, pp. 144925–144944, 2019.

[24] A. Damodaran, F. D. Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 13, no. 1, pp. 1–12, Feb. 2017.

[25] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A survey on malware detection using data mining techniques," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 1–40, May 2018.

[26] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, pp. 1–22, 2018.

[27] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, and M. Rajarajan, ''Android security: A survey of issues, malware penetration, and defenses,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 998–1022, 2nd Quart., 2015.

[28] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, ''The evolution of Android malware and Android analysis techniques,'' *ACM Comput. Surv.*, vol. 49, no. 4, pp. 1–41, Jan. 2017.

[29] D. Ucci, L. Aniello, and R. Baldoni, ''Survey of machine learning techniques for malware analysis,'' *Comput. Secur.*, vol. 81, pp. 123–147, Mar. 2019.

[30] S. Poudyal and D. Dasgupta, ''Analysis of crypto-ransomware using ML-based multi-level profiling,'' *IEEE Access*, vol. 9, pp. 122532–122547, 2021.

[31] X. Zhang, J. Wang, and S. Zhu, ''Dual generative adversarial networks based unknown encryption ransomware attack detection,'' *IEEE Access*, vol. 10, pp. 900–913, 2022.

[32] S. H. Kok, A. Abdullah, and N. Jhanjhi, ''Early detection of crypto-ransomware using pre-encryption detection algorithm,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1984–1999, May 2022.

[33] S. H. Kok, A. Azween, and N. Jhanjhi, ''Evaluation metric for crypto-ransomware detection using machine learning,'' *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102646.

[34] J. A. Gómez-Hernández, L. Álvarez-González, and P. García-Teodoro, ''R-Locker: Thwarting ransomware action through a honeyfile-based approach,'' *Comput. Secur.*, vol. 73, pp. 389–398, Mar. 2018.

[35] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, and F. Martinelli, ''R-PackDroid: API package-based characterization and detection of mobile ransomware,'' in *Proc. Symp. Appl. Comput.*, Apr. 2017, pp. 1718–1723.

[36] R. Moussaileb, N. Cuppens, J. L. Lanet, and H. Le Bouder, ''Ransomware network traffic analysis for pre-encryption alert,'' in *Ransomware Network Traffic Analysis for Pre-Encryption Alert* (Lecture Notes in Computer Science), vol. 12056. Cham, Switzerland: Springer, 2020.

[37] M. A. Taylor, E. C. Larson, and M. A. Thornton, ''Rapid ransomware detection through side channel exploitation,'' in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2021, pp. 47–54.

[38] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, C. Mulliner, and W. Robertson, ''UNVEIL: A large-scale, automated approach to detecting ransomware,'' in *Proc. USENIX Secur. Symp.*, 2016, pp. 1–17.

[39] J. Zhang, Z. Gu, J. Jang, D. Kirat, M. Stoecklin, X. Shu, and H. Huang, ''Scarecrow: Deactivating evasive malware via its own evasive logic,'' in *Proc. 50th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2020, pp. 76–87.

[40] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, ''Asynchronous peer-to-peer federated capability-based targeted ransomware detection model for industrial IoT,'' *IEEE Access*, vol. 9, pp. 148738–148755, 2021.

[41] A. Gharib and A. Ghorbani, ''DNA-Droid: A real-time Android ransomware detection framework,'' in *Network and System Security* (Lecture Notes in Computer Science), Cham, Switzerland: Springer, vol. 10394. 2017.

[42] Z.-G. Chen, H.-S. Kang, S.-N. Yin, and S.-R. Kim, ''Automatic ransomware detection and analysis based on dynamic API calls flow graph,'' in *Proc. Int. Conf. Res. Adapt. Convergent Syst.*, Sep. 2017, pp. 196–201.

[43] S. Chadha and U. Kumar, ''Ransomware: Let's fight back!'' in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 925–930.

[44] C. Zheng, N. Dellarocca, N. Andronio, S. Zanero, and F. Maggi, ''GreatEatlon: Fast, static detection of mobile ransomware,'' in *Security and Privacy in Communication Networks* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering). Cham, Switzerland: Springer, 2017.

[45] R. Vinayakumar, K. P. Soman, K. K. Senthil Velan, and S. Ganorkar, ''Evaluating shallow and deep networks for ransomware detection and classification,'' in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 259–265.

[46] G. Cusack, O. Michel, and E. Keller, ''Machine learning-based detection of ransomware using SDN,'' in *Proc. ACM Int. Workshop Secur. Softw. Defined Netw. Netw. Function Virtualization*, Mar. 2018, pp. 1–6.

[47] M. M. Hasan and M. M. Rahman, ''RansHunt: A support vector machines based ransomware analysis framework with integrated feature set,'' in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2017, pp. 1–7.

[48] S. Baek, Y. Jung, A. Mohaisen, S. Lee, and D. Nyang, ''SSD-Insider: Internal defense of solid-state drive against ransomware with perfect data recovery,'' in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 875–884.

[49] S. Mehnaz, A. Mudgerikar, and E. Bertino, ''RWGuard: A real-time detection system against cryptographic ransomware,'' in *Research in Attacks, Intrusions, and Defenses* (Lecture Notes in Computer Science), vol. 11050. Cham, Switzerland: Springer, 2018.

[50] A. Cohen and N. Nissim, ''Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory,'' *Exp. Syst. Appl.*, vol. 102, pp. 158–178, Jul. 2018.

[51] A. Azmoodeh, A. Dehghantanha, M. Conti, and K.-K. R. Choo, ''Detecting crypto-ransomware in IoT networks based on energy consumption footprint,'' *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 4, pp. 1141–1152, 2018.

[52] Y. Takeuchi, K. Sakai, and S. Fukumoto, ''Detecting ransomware using support vector machines,'' in *Proc. 47th Int. Conf. Parallel Process. Companion*, Aug. 2018, pp. 1–6.

[53] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, ''Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection,'' *Future Gener. Comput. Syst.*, vol. 101, pp. 476–491, Dec. 2019.

[54] N. Naik, P. Jenkins, and N. Savage, ''A ransomware detection method using fuzzy hashing for mitigating the risk of occlusion of information systems,'' in *Proc. Int. Symp. Syst. Eng. (ISSE)*, Oct. 2019, pp. 1–6.

[55] K. Lee, S.-Y. Lee, and K. Yim, ''Machine learning based file entropy analysis for ransomware detection in backup systems,'' *IEEE Access*, vol. 7, pp. 110205–110215, 2019.

[56] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, ''Classification of ransomware families with machine learning based on N-gram of opcodes,'' *Future Gener. Comput. Syst.*, vol. 90, pp. 211–221, Jan. 2019.

[57] T. Lam and H. Kettani, ''PhAttApp: A phishing attack detection application,'' in *Proc. 3rd Int. Conf. Inf. Syst. Data Mining*, Apr. 2019, pp. 154–158.

[58] S. Alsoghyer and I. Almomani, ''Ransomware detection system for Android applications,'' *Electronics*, vol. 8, no. 8, p. 868, Aug. 2019.

[59] S. Poudyal, D. Dasgupta, Z. Akhtar, and K. Gupta, ''A multi-level ransomware detection framework using natural language processing and machine learning,'' in *Proc. 14th Int. Conf. Malicious Unwanted Softw.*, 2019, pp. 1–8.

[60] H. Zuhair and A. Selamat, ''An intelligent and real-time ransomware detection tool using machine learning algorithm,'' *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 23, pp. 3448–3461, 2019.

[61] S. Poudyal, K. P. Subedi, and D. Dasgupta, ''A framework for analyzing ransomware using machine learning,'' in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1692–1699.

[62] R. Agrawal, J. W. Stokes, K. Selvaraj, and M. Marinescu, ''Attention in recurrent neural networks for ransomware detection,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3222–3226.

[63] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, and A. Chattopadhyay, ''RATAFIA: Ransomware analysis using time and frequency informed autoencoders,'' in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2019, pp. 218–227.

[64] S. I. Bae, G. B. Lee, and E. G. Im, ''Ransomware detection using machine learning algorithms,'' *Concurrency Comput., Pract. Exper.*, vol. 32, no. 18, p. e5422, 2020.

[65] A. Alqahtani, M. Gazzan, and F. T. Sheldon, ''A proposed crypto-ransomware early detection (CRED) model using an integrated deep learning and vector space model approach,'' in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2020, pp. 0275–0279.

[66] B. Qin, Y. Wang, and C. Ma, ''API call based ransomware dynamic detection approach using TextCNN,'' in *Proc. Int. Conf. Big Data, Artif. Intell. Internet Things Eng. (ICBAIE)*, Jun. 2020, pp. 162–166.

[67] J. Hwang, J. Kim, S. Lee, and K. Kim, ''Two-stage ransomware detection using dynamic analysis and machine learning techniques,'' *Wireless Pers. Commun.*, vol. 112, no. 4, pp. 2597–2609, Jun. 2020.

[68] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, ''Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence,'' *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 341–351, Apr. 2020.

[69] B. M. Khammas, "Ransomware detection using random forest technique," *ICT Exp.*, vol. 6, no. 4, pp. 325–331, Dec. 2020.

[70] M. M. J. Mary, S. Usharani, P. M. Bala, and S. G. Sandhya, "Detection of ransomware in static analysis by using gradient tree boosting algorithm," in *Proc. Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Jul. 2020, pp. 1–5.

[71] B. A. S. Al-Rimy, M. A. Maarof, M. Alazab, F. Alsolami, S. Z. M. Shaid, F. A. Ghaleb, T. Al-Hadhrami, and A. M. Ali, "A pseudo feedback-based annotated TF-IDF technique for dynamic crypto-ransomware pre-encryption boundary delineation and features extraction," *IEEE Access*, vol. 8, pp. 140586–140598, 2020.

[72] K. C. Roy and Q. Chen, "DeepRan: Attention-based BiLSTM and CRF for ransomware early detection and classification," *Inf. Syst. Frontiers*, vol. 23, no. 2, pp. 299–315, Apr. 2021.

[73] M. A. Ayub, A. Continella, and A. Siraj, "An I/O request packet (IRP) driven effective ransomware detection scheme using artificial neural network," in *Proc. IEEE 21st Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Aug. 2020, pp. 319–324.

[74] F. Manavi and A. Hamzeh, "A new method for ransomware detection based on PE header using convolutional neural networks," in *Proc. 17th Int. ISC Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2020, pp. 82–87.

[75] Y. A. Ahmed, B. Koçer, S. Huda, B. A. S. Al-rimy, and M. M. Hassan, "A system call refinement-based enhanced minimum redundancy maximum relevance method for ransomware early detection," *J. Netw. Comput. Appl.*, vol. 167, Oct. 2020, Art. no. 102753.

[76] I. Almomani, A. AlKhayer, and M. Ahmed, "An efficient machine learning-based approach for Android v.11 ransomware detection," in *Proc. 1st Int. Conf. Artif. Intell. Data Analytics (CAIDA)*, Apr. 2021, pp. 240–244.

[77] M. Almousa, S. Basavaraju, and M. Anwar, "API-based ransomware detection using machine learning-based threat detection models," in *Proc. 18th Int. Conf. Privacy, Secur. Trust (PST)*, Dec. 2021, pp. 1–7.

[78] R. M. A. Molina, S. Torabi, K. Sarieddine, E. Bou-Harb, N. Bouguila, and C. Assi, "On ransomware family attribution using pre-attack paranoia activities," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 1, pp. 19–36, Mar. 2022.

[79] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic," *Exp. Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118299.

[80] S. Kumar, "An effective ransomware detection approach in a cloud environment using volatile memory features," *J. Comput. Virol. Hacking Techn.*, vol. 18, no. 4, pp. 407–424, Apr. 2022.

[81] A. Singh, R. Ikuesan, and H. Venter, "Ransomware detection using process memory," in *Proc. Int. Conf. Cyber Warfare Secur.*, 2022, vol. 17, no. 1, pp. 1–10.

[82] Y. A. Ahmed, S. Huda, B. A. S. Al-rimy, N. Alharbi, F. Saeed, F. A. Ghaleb, and I. M. Ali, "A weighted minimum redundancy maximum relevance technique for ransomware early detection in industrial IoT," *Sustainability*, vol. 14, no. 3, p. 1231, Jan. 2022.

[83] Z. Li, A. L. G. Rios, and L. Trajkovic, "Machine learning for detecting the WestRock ransomware attack using BGP routing records," *IEEE Commun. Mag.*, vol. 61, no. 3, pp. 20–26, Mar. 2023.

[84] A. Azman, W. Yassin, O. Mohd, M. F. Abdollah, and R. S. Abdullah, "Ransomware detection using classification method against registry data," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 22, pp. 3366–3376, 2019.

[85] H. Faris, M. Habib, I. Almomani, M. Eshtay, and I. Aljarah, "Optimizing extreme learning machines using chains of salps for efficient Android ransomware detection," *Appl. Sci.*, vol. 10, no. 11, p. 3706, May 2020.

[86] Z. Abdullah, F. W. Muhadi, M. M. Saudi, I. R. A. Hamid, and C. F. M. Foozy, "Android ransomware detection based on dynamic obtained features," in *Recent Advances on Soft Computing and Data Mining* (Advances in Intelligent Systems and Computing), vol. 978. Cham, Switzerland: Springer, 2020.

[87] A. N. Jahromi, S. Hashemi, A. Dehghantanha, K.-K.-R. Choo, H. Karimipour, D. E. Newton, and R. M. Parizi, "An improved two-hidden-layer extreme learning machine for malware hunting," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101655.

[88] D. T. Nguyen and S. Lee, "LightGBM-based ransomware detection using API call sequences," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 10, pp. 138–146, 2021.

[89] M. Basnet, S. Poudyal, M. H. Ali, and D. Dasgupta, "Ransomware detection using deep learning in the SCADA system of electric vehicle charging station," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf.-Latin Amer.*, Sep. 2021, pp. 1–5.

[90] C.-M. Hsu, C.-C. Yang, H.-H. Cheng, P. E. Setiasabda, and J.-S. Leu, "Enhancing file entropy analysis to improve machine learning detection rate of ransomware," *IEEE Access*, vol. 9, pp. 138345–138351, 2021.

[91] M. Almousa, J. Osawere, and M. Anwar, "Identification of ransomware families by analyzing network traffic using machine learning techniques," in *Proc. 3rd Int. Conf. Transdisciplinary AI (TransAI)*, Sep. 2021, pp. 19–24.

[92] N. Aljubory and B. M. Khammas, "Hybrid evolutionary approach in feature vector for ransomware detection," in *Proc. Int. Conf. Intell. Technol., Syst. Service Internet Everything (ITSS-IoE)*, Nov. 2021, pp. 1–6.

[93] A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barenghi, S. Zanero, and F. Maggi, "ShieldFS: A self-healing, ransomware-aware filesystem," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl.*, Dec. 2016, pp. 336–347.

[94] I. Almomani, R. Qaddoura, M. Habib, S. Alsoghyer, A. A. Khayer, I. Aljarah, and H. Faris, "Android ransomware detection based on a hybrid evolutionary approach in the context of highly imbalanced data," *IEEE Access*, vol. 9, pp. 57674–57691, 2021.

[95] B. A. S. Al-rimy, M. A. Maarof, M. Alazab, S. Z. M. Shaid, F. A. Ghaleb, A. Almalawi, A. M. Ali, and T. Al-Hadhrami, "Redundancy coefficient gradual Up-weighting-based mutual information feature selection technique for crypto-ransomware early detection," *Future Gener. Comput. Syst.*, vol. 115, pp. 641–658, Feb. 2021.

[96] M. E. Ahmed, H. Kim, S. Camtepe, and S. Nepal, "Peeler: Profiling kernel-level events to detect ransomware," in *Computer Security—ESORICS* (Lecture Notes in Computer Science), vol. 12972. Cham, Switzerland: Springer, 2021.

[97] M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, "Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares," in *Proc. 12th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2015, pp. 79–84.

[98] S. Jung and Y. Won, "Ransomware detection method based on context-aware entropy analysis," *Soft Comput.*, vol. 22, no. 20, pp. 6731–6740, Oct. 2018.

[99] S. S. Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi, and B. Raman, "Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks," *IEEE Access*, vol. 8, pp. 169944–169956, 2020.

[100] A. O. Almashhadani, M. Kaiiali, S. Sezer, and P. O'Kane, "A multi-classifier network-based crypto ransomware detection system: A case study of Locky ransomware," *IEEE Access*, vol. 7, pp. 47053–47067, 2019.

[101] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and drop It): Stopping ransomware attacks on user data," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 303–312.

[102] C. Moore, "Detecting ransomware with honeypot techniques," in *Proc. Cybersecurity Cyberforensics Conf. (CCC)*, Aug. 2016, pp. 77–81.

[103] A. Zahra and M. A. Shah, "IoT based ransomware growth rate evaluation and detection using command and control blacklisting," in *Proc. 23rd Int. Conf. Autom. Comput. (ICAC)*, Sep. 2017, pp. 1–6.

[104] A. Alzahrani, A. Alshehri, H. Alshahrani, R. Alharthi, H. Fu, A. Liu, and Y. Zhu, "RanDroid: Structural similarity approach for detecting ransomware applications in Android platform," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, May 2018, pp. 0892–0897.

[105] D. Min, D. Park, J. Ahn, R. Walker, J. Lee, S. Park, and Y. Kim, "Amoeba: An autonomous backup and recovery SSD for ransomware attack defense," *IEEE Comput. Archit. Lett.*, vol. 17, no. 2, pp. 245–248, Jul. 2018.

[106] D. Morato, E. Berrueta, E. Magaña, and M. Izal, "Ransomware early detection by the analysis of file sharing traffic," *J. Netw. Comput. Appl.*, vol. 124, pp. 14–32, Dec. 2018.

[107] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," *Comput. Electr. Eng.*, vol. 66, pp. 353–368, Feb. 2018.

[108] K. P. Subedi, D. R. Budhathoki, and D. Dasgupta, "Forensic analysis of ransomware families using static and dynamic analysis," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 180–185.

[109] S. Salehi, H. Shahriari, M. M. Ahmadian, and L. Tazik, "A novel approach for detecting DGA-based ransomwares," in *Proc. ISCISC*, Aug. 2018, pp. 1–7.

[110] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," *Comput. Secur.*, vol. 73, pp. 326–344, Mar. 2018.

[111] M. Junaid, J. Ming, and D. Kung, "StateDroid: Stateful detection of stealthy attacks in Android apps via horn-clause verification," in *Proc. ACM Int. Conf. Proc. Ser.*, Jan. 2018, pp. 198–209.

[112] A. Singh, A. Ikuesan, and H. Venter, "A context-aware trigger mechanism for ransomware forensics," in *Proc. 14th Int. Conf. Cyber Warfare Secur.*, 2019, p. 629.

[113] N. Naik, P. Jenkins, N. Savage, L. Yang, K. Naik, and J. Song, "Augmented YARA rules fused with fuzzy hashing in ransomware triaging," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 625–632.

[114] G. Ramesh and A. Menen, "Automated dynamic approach for detecting ransomware using finite-state machine," *Decis. Support Syst.*, vol. 138, Nov. 2020, Art. no. 113400.

[115] S. Wang, H. Zhang, S. Qin, W. Li, T. Tu, A. Shen, and W. Liu, "KRProtector: Detection and files protection for IoT devices on Android without ROOT against ransomware based on decoys," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18251–18266, Oct. 2022.

[116] J. A. Gómez-Hernández, R. Sánchez-Fernández, and P. García-Teodoro, "Inhibiting crypto-ransomware on windows platforms through a honeyfile-based approach with R-Locker," *IET Inf. Secur.*, vol. 16, no. 1, pp. 64–74, Jan. 2022.

[117] S. Maniath, A. Ashok, P. Poornachandran, V. G. Sujadevi, A. U. Sankar, and S. Jan, "Deep learning LSTM based ransomware detection," in *Proc. Recent Develop. Control, Automat. Power Eng. (RDCAPE)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, May 2018, pp. 442–446.

[118] H. Daku, P. Zavarsky, and Y. Malik, "Behavioral-based classification and identification of ransomware variants using machine learning," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1560–1564.

[119] R. Vinayakumar, M. Alazab, A. Jolfaei, K. P. Soman, and P. Poornachandran, "Ransomware triage using deep learning: Twitter as a case study," in *Proc. Cybersecurity Cyberforensics Conf. (CCC)*, May 2019, pp. 67–73.

[120] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K. K. R. Choo, and D. E. Newton, "DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer," *Future Gener. Comput. Syst.*, vol. 90, pp. 94–104, Jan. 2019.

[121] S. Usharani, P. M. Bala, and M. M. J. Mary, "Dynamic analysis on crypto-ransomware by using machine learning: GandCrab ransomware," *J. Phys., Conf.*, vol. 1717, no. 1, Jan. 2021, Art. no. 012024.

[122] A. Wani and S. Revathi, "Ransomware protection in IoT using software defined networking," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 10, no. 3, p. 3166, Jun. 2020.

[123] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2018, pp. 356–363.

[124] A. Kharraz and E. Kirda, "Redemption: Real-time protection against ransomware at end-hosts," in *Research in Attacks, Intrusions, and Defenses* (Lecture Notes in Computer Science), vol. 10453. Cham, Switzerland: Springer 2017.

[125] S. Song, B. Kim, and S. Lee, "The effective ransomware prevention technique using process monitoring on Android platform," *Mobile Inf. Syst.*, vol. 2016, pp. 1–9, Jan. 2016.

[126] S. Lee, H. K. Kim, and K. Kim, "Ransomware protection using the moving target defense perspective," *Comput. Electr. Eng.*, vol. 78, pp. 288–299, Sep. 2019.

[127] J. K. Lee, S. Y. Moon, and J. H. Park, "CloudRPS: A cloud analysis based enhanced ransomware prevention system," *J. Supercomput.*, vol. 73, no. 7, pp. 3065–3084, Jul. 2017.

[128] O. Ami, Y. Elovici, and D. Hendler, "Ransomware prevention using application authentication-based file access control," in *Proc. 33rd Annu. ACM Symp. Appl. Comput.*, Apr. 2018, pp. 1610–1619.

[129] A. Alsabeh, H. Safa, E. Bou-Harb, and J. Crichigno, "Exploiting ransomware paranoia for execution prevention," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2020, pp. 1–6.

[130] H. Kim, D. Yoo, J.-S. Kang, and Y. Yeom, "Dynamic ransomware protection using deterministic random bit generator," in *Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS)*, Nov. 2017, pp. 64–68.

[131] F. Faghihi and M. Zulkernine, "RansomCare: Data-centric detection and mitigation against smartphone crypto-ransomware," *Comput. Netw.*, vol. 191, May 2021, Art. no. 108011.

[132] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, "PayBreak: Defense against cryptographic ransomware," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 599–611.

[133] J. S. Aidan and U. Garg, "Advanced Petya ransomware and mitigation strategies," in *Proc. 1st Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, Dec. 2018, pp. 23–28.

[134] K. Cabaj and W. Mazurczyk, "Using software-defined networking for ransomware mitigation: The case of CryptoWall," *IEEE Netw.*, vol. 30, no. 6, pp. 14–20, Nov. 2016.

[135] E. Rouka, C. Birkinshaw, and V. G. Vassilakis, "SDN-based malware detection and mitigation: The case of ExPetr ransomware," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 150–155.

[136] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Ransomware detection and mitigation using software-defined networking: The case of WannaCry," *Comput. Electr. Eng.*, vol. 76, pp. 111–121, Jun. 2019.

[137] M. A. S. Monge, J. M. Vidal, and L. J. G. Villalba, "A novel self-organizing network solution towards crypto-ransomware mitigation," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, Aug. 2018, pp. 1–10.

[138] L. Fernández Maimó, A. Huertas Celdrán, Á. Perales Gómez, F. García Clemente, J. Weimer, and I. Lee, "Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments," *Sensors*, vol. 19, no. 5, p. 1114, Mar. 2019.

[139] S. R. Davies, R. Macfarlane, and W. J. Buchanan, "Evaluation of live forensic techniques in ransomware attack mitigation," *Forensic Sci. Int., Digit. Invest.*, vol. 33, Jun. 2020, Art. no. 300979.

[140] R. Umar, I. Riadi, and R. S. Kusuma, "Mitigating sodinokibi ransomware attack on cloud network using software-defined networking (SDN)," *Int. J. Saf. Secur. Eng.*, vol. 11, no. 3, pp. 239–246, Jun. 2021.

[141] V. Mathane and P. V. Lakshmi, "Predictive analysis of ransomware attacks using context-aware AI in IoT systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, pp. 240–244, 2021.

[142] C. G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, "BitcoinHeist: Topological data analysis for ransomware prediction on the Bitcoin blockchain," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4439–4445.

[143] M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," *Comput. Secur.*, vol. 77, pp. 578–594, Aug. 2018.

[144] S. Xu, "The application of machine learning in Bitcoin ransomware family prediction," in *Proc. 5th Int. Conf. Inf. Syst. Data Mining*, May 2021, pp. 21–27.

[145] H.-Y. Chang, T.-L. Lin, T.-F. Hsu, Y.-S. Shen, and G.-R. Li, "Implementation of ransomware prediction system based on weighted-KNN and real-time isolation architecture on SDN networks," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2019, pp. 1–2.

[146] F. Quinkert, H. Thorsten, K. Hossain, F. Emilio, and L. Kristina, "RAPTOR: Ransomware attack PredicTOR," 2018, *arXiv:1803.01598*.

[147] U. Adamu and I. Awan, "Ransomware prediction using supervised learning algorithms," in *Proc. 7th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2019, pp. 57–63.

[148] L. Chen, Y. Ye, and T. Bourlai, "Adversarial machine learning in malware detection: Arms race between evasion attack and defense," in *Proc. Eur. Intell. Secur. Informat. Conf. (EISIC)*, Sep. 2017, pp. 99–106.

[149] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," in *Proc. 26th Eur. Signal Process. Conf.*, Sep. 2018, pp. 533–537.

[150] X. Chen, C. Li, D. Wang, S. Wen, J. Zhang, S. Nepal, Y. Xiang, and K. Ren, "Android HIV: A study of repackaging malware for evading machine-learning detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, no. 1, pp. 987–1001, Jul. 2019.

SALWA RAZAULLA received the bachelor's degree in information technology and the master's degree in computer science engineering from Osmania University, India. She is currently a Research Assistant with the College of Engineering and IT, University of Dubai. She is also a Cybersecurity Researcher. Her research interests include cybersecurity, including network security, cryptography, and machine learning. She is particularly interested in exploring the applications of machine learning in responding to various threats.

CLAUDE FACHKHA received the B.Eng. degree in computer and communication from the University of Notre Dame, in 2008, and the master's degree in information systems security engineering and the Ph.D. degree in electrical and computer engineering from Concordia University, Canada, in 2010 and 2015, respectively. He is currently an Assistant Professor with the College of Engineering and IT, University of Dubai. He is also the Co-Founder of Steppa Cyber Inc., Canada. His research interests include cyber security, data science, the IoT, data mining, and machine learning. He was a recipient of the prestigious Fonds de Recherche du Quebec–Nature et Technologies (FQRNT) Award from Canada. He served as a Technical Editor for *IEEE Communications Magazine*.

CHRISTINE MARKARIAN (Member, IEEE) received the B.S. degree in mathematics from Haigazian University, Lebanon, the M.Sc. degree in computer science from Lebanese American University, Lebanon, and the Ph.D. degree in computer science from Paderborn University, Germany. She is currently an Assistant Professor of computer science with the College of Engineering and IT Department, University of Dubai. Her research interests include combinatorial optimization, design and analysis of algorithms, online algorithms, operations research, distributed computing, cloud computing, wireless networks, robotics, and cyber security. She is a member of *Mathematical Reviews*. She served/serves as a Reviewer for several prestigious journals and conferences, including *Ad Hoc Networks*, *Theoretical Computer Science*, the Symposium on Parallelism in Algorithms and Architectures (SPAA), the International Conference on Intelligent Robots and Systems (IROS 2017), *Swarm Intelligence* (SI), the International Symposium on Mathematical Foundations of Computer Science (MFCS), the Algorithms and Data Structures Symposium (WADS), the Annual ACM Symposium on Principles of Distributed Computing (PODC), the International Colloquium on Structural Information and Communication Complexity (SIROCCO), and the Conference on Algorithms and Complexity (CIAC).

AMJAD GAWANMEH (Senior Member, IEEE) received the bachelor's degree in electrical and computer engineering from JUST, Jordan, in 1998, and the M.S. and Ph.D. degrees from Concordia University, Montreal, Canada, in 2003 and 2008, respectively. He is currently an Associate Professor with the University of Dubai. He is also the Director of the Electrical Engineering Program, United Arab Emirates, and an Affiliate Adjunct Professor with Concordia University. He has edited two books, three book chapters, more than 40 peer-reviewed Scopus-indexed journal articles, and more than 75 peer-reviewed conference papers. He is the UAE GRSS Chapter Chair.

WATHIQ MANSOOR (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Aston University, U.K., where he focused on the design and implementation of multiprocessor systems and communication protocols for computer vision applications. He is currently an Esteemed Professor with the University of Dubai with extensive academic leadership experience in renowned universities worldwide. He has supervised numerous Ph.D. and undergraduate projects in the field of computer engineering and innovation in business. In addition, he has co-supervised several postgraduate students through research collaboration with international research groups. Overall, his contributions to the field of computer engineering and artificial intelligence are remarkable and continue to inspire students and professionals alike. He is an accomplished organizer of international and national conferences and workshops. He has published more than 180 journals and conference papers in the field of computer engineering in the last seven years, with a specific emphasis on AI. His current research interests include artificial intelligence, intelligent systems, security, and utilizing neural networks with deep learning models for various applications. He is a Senior Member of the IEEE UAE Section.

BENJAMIN C. M. FUNG (Senior Member, IEEE) received the Ph.D. degree in computing science from Simon Fraser University, Canada, in 2007. He is currently the Canada Research Chair of data mining for cybersecurity and a Professor with the School of Information Studies, McGill University, Canada. He has more than 130 refereed publications, with more than 11,000 citations, that span the research forums of data mining, privacy protection, cybersecurity, services computing, and building engineering. He serves as an Associate Editor for IEEE Transactions on Knowledge and Data Engineering and *Sustainable Cities and Society* (Elsevier). He is also a licensed Professional Engineer of software engineering in the Province of Ontario, Canada.

CHADI ASSI (Fellow, IEEE) received the B.Eng. degree from Lebanese University, Beirut, Lebanon, in 1997, and the Ph.D. degree from the Graduate Center, City University of New York, New York, NY, USA, in April 2003. He is currently a Professor with Concordia University, where he holds the Tier I University Research Chair. Before joining Concordia University, he was a Visiting Scientist for one year with Nokia Research Center, Boston, MA, USA, from 2002 to 2003, working on quality-of-service in optical access networks. He is supervising a group of 14 Ph.D. students and four M.A.Sc. students and has successfully supervised 18 Ph.D. students and 25 M.A.Sc. students. His students received very prestigious awards from NSERC and FQRNT. His current research interests include networks, network design and modeling, network optimization, resource virtualization, and network and cyber security. He received the prestigious Mina Rees Dissertation Award from the City University of New York, in August 2002, for his research on wavelength-division-multiplexing optical networks and lightpath provisioning. He was the Tier II University Chair with Concordia University, from 2012 to 2017, in the area of wireless networks. He is on the editorial board of IEEE Communications Surveys and Tutorials. He serves as an Associate Editor for IEEE Transactions on Vehicular Technology, IEEE Transactions on Communications, IEEE Transactions on Mobile Computing, and IEEE Transactions on Network and Service Management.