IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 February 2024, accepted 3 May 2024, date of publication 7 May 2024, date of current version 22 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3397921

SURVEY

Ransomware Detection Using Machine Learning: A Review, Research Limitations and Future Directions

JAMIL ISPAHANY[1,2], (Graduate Student Member, IEEE),
MD. RAFIQUL ISLAM[1,3], (Senior Member, IEEE),
MD. ZAHIDUL ISLAM[1,2], AND
M. ARIF KHAN[1,2], (Member, IEEE)

[1]Cyber Security Cooperative Research Centre (CSCRC), Kingston, ACT 2600, Australia
[2]School of Computing, Mathematics and Engineering, Charles Sturt University, Bathurst, NSW 2795, Australia
[3]School of Computing, Mathematics and Engineering, Charles Sturt University, Albury–Wodonga, Thurgoona, NSW 2640, Australia

Corresponding author: Jamil Ispahany (jispahany@csu.edu.au)

This work was supported in part by Cyber Security Research Centre Ltd., through the Australian Government's Cooperative Research Centres Program.

ABSTRACT Ransomware attacks are on the rise in terms of both frequency and impact. The shift to remote work due to the COVID-19 pandemic has led more people to work online, prompting companies to adapt quickly. Unfortunately, this increased online activity has provided cybercriminals numerous opportunities to carry out devastating attacks. One recent method employed by malicious actors involves infecting corporate networks with ransomware to extract millions of dollars in profits. Ransomware falls into the category of malware. It works by encrypting sensitive data and demanding payments from victims to receive the encryption keys necessary for decrypting their data. The prevalence of this type of attack has prompted governments and organisations worldwide to intensify their efforts to combat ransomware. In response, the research community has also focused on ransomware detection, leveraging technologies such as machine learning. Despite this increased attention, practical solutions for real-world applications remain scarce in the existing literature. Numerous surveys have explored literature within the domain. Still, there is a notable lack of emphasis on the design of ransomware detection systems and the practical aspects of detection, including real-time and early detection. Against this backdrop, our review delves into the existing literature on ransomware detection, specifically examining the machine-learning techniques, detection approaches, and designs employed. Finally, we highlight the limitations of prior studies and propose future research directions in this crucial area.

INDEX TERMS Ransomware detection, machine learning, deep learning, early detection, real-time detection, survey.

I. INTRODUCTION

Since the beginning of the COVID-19 crisis, the proliferation of malware has become a global challenge [1]. One particular strain of malware that has gained notoriety recently is ransomware, which encrypts sensitive files and sells the decryption keys back to the victim. Despite improvements in malware detection systems, ransomware attacks have increased, prompting governments to issue sanctions against organised crime groups facilitating ransomware attacks [2]. During the first half of 2021 alone, ransomware payments totalled USD 590 million, trumping $416 million worth of losses the year before [2]. Factors that have led to the dramatic increase in ransomware attacks include:

- The advent of cryptocurrencies: Cryptocurrencies have substantially increased the risk of illicit online activities,

The associate editor coordinating the review of this manuscript and approving it for publication was Sunith Bandaru.

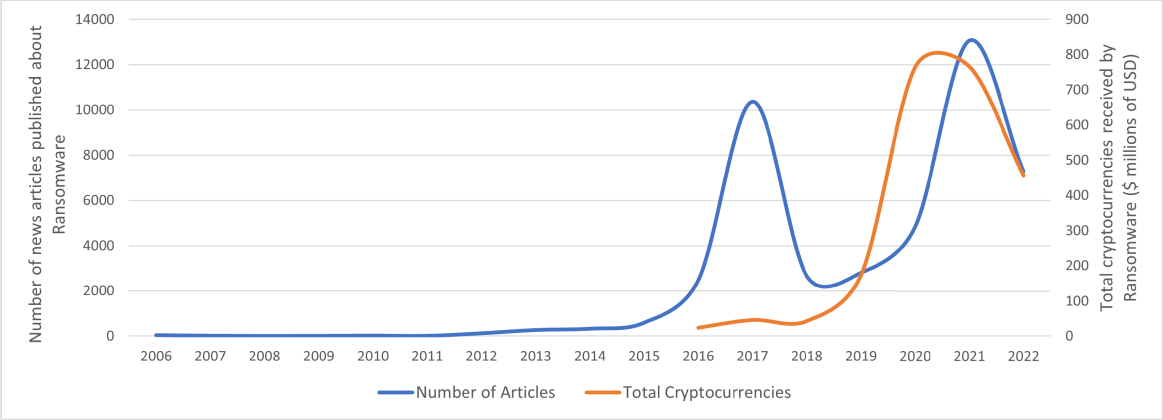

FIGURE 1. Number of global news items published about ransomware and the total cryptocurrencies received into ransomware addresses in millions [3].

enabling malicious actors to launch attacks globally while concealing their identities. This was highlighted by the cybercriminal group REvil, which accumulated over $200 million in ransomware payments in Bitcoin and Monero from companies across the globe [2].

- More people are working online: Shortly after the COVID-19 pandemic began, there was a dramatic increase in people connecting to the internet because of lockdowns [4]. This forced companies worldwide to shift their workforce online to continue business operations. In response, organisations adapted rapidly, transforming how business is conducted online [5]. As a result, this has increased the attack surface for criminals, ultimately creating more avenues for malicious actors to launch attacks.
- The rise of criminal cooperatives: Cybercriminals no longer work in independent silos. Online criminal organisations have started collaborating for economies of scale. Modern malware is often an amalgamation of malicious scripts or executables written by several parties. For example, one party may focus on creating the ransomware executable while another may specialise in breaching the target network and harvesting sensitive information [6]. This has led to business models such as Ransomware-as-a-Service (RaaS)[1] which offers affiliates commissions for distribution.

These factors have created the perfect environment for malicious actors to launch devastating ransomware attacks, which are becoming bolder and more frequent, as seen during the attack on Colonial Pipeline Co., the biggest oil pipeline company in the United States (U.S.). On 7 May 2021, the hacking group Darkside breached the company's network via leaked credentials from the dark web. Once inside, the hackers exfiltrated 100 gigabytes of company data and released ransomware across the network before demanding USD 4.4 million to restore files and cease the attack [7]. To mitigate further damage, the company shut down operations for several days, resulting in supply and demand constraints across the East Coast, inflating the gas price to more than $3 per gallon [8]. Despite recommendations against submitting to ransom demands, the company's CEO released USD 4.4 million worth of cryptocurrencies to the Darkside group, of which $2.3 million was later recovered by the Department of Justice from crypto wallets used during the exchange.

The Costa Rican government suffered a similar attack in 2022 from the ransomware group Conti, who demanded $20 million to recover critical files after releasing ransomware across various departments. As a result of the attack, international trade within the country ground to a halt, and over 1,500 servers were infected, forcing government staff to revert to pen and paper to complete operations. A national emergency was declared to mitigate further damages, and although the ransom was not paid, businesses suffered losses of up to $125 million during 48 hours [9].

In response to the growing threat, governments worldwide are pursuing legal avenues [2] to increase security controls in major organisations as shown in Table 1. As a result, many countries are enforcing mandatory reporting requirements for organisations managing critical infrastructures such as gas pipelines, electricity distributors and hospitals. By design, many of these laws increase attack transparency and encourage victims to respond to security incidents promptly, ultimately reducing the impact on the public. While organisations are advised not to pay the ransom after an attack, governments often fall short in outlawing the payment to malicious actors to retrieve their data. The U.S. has led legal efforts to disrupt ransomware groups, such as establishing sanctions against ransomware operators and designating several virtual currency exchanges as complicit

[1]Similar to the Software-as-a-Service model, the ransomware backend infrastructure is managed by the malicious actors and distributed by affiliates for a commission.

[2]https://www.whitehouse.gov/briefing-room/statements-releases/2021/10/14/joint-statement-of-the-ministers-and-representatives-from-the-counter-ransomware-initiative-meeting-october-2021/

TABLE 1. Various acts and bills by governments and worldwide to tackle the growing threat of Ransomware. [n] A bill or act was proposed but did not proceed.

Country	Document	Description
Australia	Security of Critical Infrastructure Act 2018	Mandatory reporting requirements for Critical infrastructure
	Crimes Legislation Amendment (Ransomware Action Plan) Bill 2022 [n]	Describes offences involving extortion of data, handling of unauthorised data, and seizing of digital assets. Offences range from 25 years imprisonment for attacks against critical infrastructure, 10 years for cyber extortion, 10 years for producing, supplying or obtaining data under an arrangement for payment
	Ransomware Payments Bill 2021 [n]	Mandatory reporting requirements for entities that make ransomware payments
Brazil	law 14.155 of the Brazilian Penal Code	Stricter laws for unauthorised tampering with data with up to 5 years' imprisonment for electronic damage
Canada	BILL C-26	Amendments to the Telecommunications Act, the establishment of Cyber Security programs, reporting of Cyber Incidents
European Union	Network and Information Security Directive (NIS2 Directive)	An update to the NIS directive to improve supply chain risk, improve reporting requirements and stricter enforcement requirements
	General Data Protection Regulation (GDPR)	Mandatory controls to protect sensitive data
India	The Information Telecommunication Act of 2000 (70B)	Mandatory reporting for cyber incidents within 6 hours.
Saudi Arabia	Personal Data Protection Law (PDPL)	Laws designed to protect and regulate the collection, processing, disclosure, or retention of personal data.
U.K	The Network and Information Systems Regulations 2018	Mandatory reporting for operators of essential services. Fines of up to 17 million pounds for not establishing effective cyber security measures
U.S	Cyber Incident Reporting Act of 2021	Mandatory reporting for ransomware payments made by critical infrastructure operators (within 24 hours for companies with more than 50 employees)
	Cyber Incident Notification Act of 2021	Mandatory reporting for cyber intrusions within 24 hours for covered entities
	Sanction and Stop Ransomware Act of 2021	Designations of state sponsors, imposing of sanctions, mandatory reporting requirements for ransomware attacks and payments, regulation of cryptocurrency exchanges
	Ransomware and Financial Stability Act of 2021	Forbids financial institutions from making ransomware payments without federal approval
	Ransom Disclosure Act	Mandatory reporting for ransomware payments to the Department of Homeland Security within 48 hours of paying a ransom

in facilitating financial transactions for ransomware actors. These efforts have led to a dramatic decline in revenue generated by ransomware, as shown in Fig. 1. Insurance companies offering cyber extortion policies also play an essential role in mitigating the ransomware threat. In most cases, subscribing to these policies requires organisations to adhere to a minimum standard of security controls to reduce the attack surface. This approach, however, can encourage ransomware groups to demand higher ransoms since insurance companies cover the cost [10].

The research community has increased their focus on ransomware detection using machine learning (ML) to mitigate the rise of ransomware attacks. The earliest known study containing the keywords "ransomware detection" within Scopus appeared in 2016.[3] This aligns with the rise in ransomware-related News items published [4] worldwide due to increased ransomware attacks in 2016 as shown in Fig. 1. Several surveys have reviewed the ransomware detection domain over this period as shown in Table. 3. These surveys broadly focus on: (1) the features or input data used to train machine learning algorithms [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32]; (2) ransomware behaviour and trends [13], [14], [16], [17], [18], [19], [20], [21], [22], [25], [27], [28], [31], [32]; (3) detection techniques [11], [12], [13], [14], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [29], [30], [31], [32]; (4) the algorithms used to detect ransomware [11], [12], [13], [15], [16], [18], [19], [25], [26], [27], [28], [29], [30], [31], [32]; (5) ransomware prevention strategies [20]. However, despite recent research efforts, several key limitations emerge from existing surveys:

1) Previously presented ransomware trends have become outdated. Ransomware trends have been covered in several surveys [13], [14], [16], [17], [18], [19], [20], [21], [22], [25], [27], [31], [32]. Still, the trends covered in previous surveys need to be updated due to the evolving nature of ransomware. Razaulla et al. [27] point out the need for more attention to the top ransomware, earning the highest revenue over the past few years.
2) The design of ransomware detection systems is understudied. No other survey holistically reviews the design approaches of existing ransomware detection systems through a pragmatic lens. Such an approach is important to determine if the proposals presented throughout the literature are suitable for "real-world" use within enterprise environments. Urooj et al. [31] reviewed some real-time detection systems throughout the literature and highlighted the lack of practicality of current proposals. However, little explanation is given as to why. Only Or-Meir et al. [24] provided a

[3]https://www.scopus.com/
[4]https://www.dowjones.com/products/factiva

comprehensive review of the design aspects of malware analysis environments. However, since their study only focuses on analysis techniques, it does not adequately explore ransomware detection using machine learning.

3) Real-time and early detection techniques are neglected. Despite being an important facet, no other studies have examined the mechanics of real-time and early detection approaches in-depth or explained what differentiates them. A handful of studies have attempted to cover early and real-time detection in their work. Consequently, the terms ''Dynamic detection'', ''early detection'', and ''real-time detection'' have been used interchangeably despite being separate paradigms. An example can be seen in the survey by Mousaileb et al. [23], which groups delayed detection techniques using dynamic analysis under the ''real-time'' analysis banner. Another example is in the survey by Alqahtani and Sheldon [12]. In their study, the authors investigated the techniques to build early detection ML models and discussed their limitations. Similar to the aforementioned survey, the ''real-time'' and ''early-detection'' paradigms are grouped.
4) There is a lack of clarity around early detection techniques. The importance of detecting ransomware before files are encrypted has been highlighted in previous surveys [20]. Despite this, most surveys simplistically view early detection as the discovery of ransomware anytime before the encryption of files. As a result, ransomware researchers have little guidance on when and where features can be collected during the ransomware attack lifecycle to detect ransomware early. Alqahtani and Sheldon [12] discuss early detection and its limitations in their survey. The authors highlight the challenges of collecting features in the early phases of a ransomware attack but recommend exploring cryptographic API calls as a solution, which is very late in the attack lifecycle. Al-Rimy et al. [14], and Urooj et al. [31] highlight early ransomware detection studies throughout the literature and provide recommendations accordingly, without distinguishing between different stages of the attack lifecycle. The most comprehensive survey on early-detection techniques was conducted by Mousaileb et al. [23]. In their survey, the authors group feature types by ransomware attack phases. However, the authors neglect ML-based detection approaches during the delivery phase of an attack, failing to capitalise on early detection techniques before the ransomware reaches the host.

Given this context, our survey examines machine learning ML-based ransomware detection systems in the existing literature. The goal is to enhance the feasibility of forthcoming proposals. To achieve this objective, we systematically address the previously mentioned limitations in this survey. We take a comprehensive approach by scrutinising ransomware detection systems and exploring machine learning mechanisms, detection approaches, and architecture. Additionally, we identify crucial limitations and provide recommendations for guiding future research.

A. CONTRIBUTIONS

To address the gaps in previous surveys and direct research efforts toward improving the practicality of ransomware detection systems, we offer the following contributions:

1) Review of the latest ransomware families, trends and government responses: We review the most pervasive ransomware families seen worldwide during recent attacks in section III-B. We also comprehensively review the latest trends and the tactics, techniques and procedures (TTPs) observed in recent ransomware attacks.
2) Present a taxonomy for ransomware detection systems using ML: In section IV, we survey ransomware detection studies across the literature to understand design approaches, detection phases and ML techniques. We propose a taxonomy as shown in Fig. 7 to assist researchers in planning ransomware detection systems' design. New contributions can be seen with a red dashed outline.
3) Review early and real-time detection approaches: We shed light on the ''early-detection'' paradigm in section IV-A by aligning studies with the Cyber-Kill-chain (CKC) to determine the earliest point at which they can detect ransomware. We categorise studies according to their position on the CKC to understand the features that can be collected from different parts of the attack lifecycle. Finally, we highlight the subtleties between early and real-time detection in section IV-C1 to reduce ambiguity in future research.
4) Review ransomware detection datasets used throughout the literature: Several studies have highlighted the importance of using standardised datasets. Despite this, most studies still build their datasets from scratch using virus repositories such as VirusTotal to train and test ransomware detection systems. This becomes cumbersome for cybersecurity researchers to validate results with other studies since the datasets differ between studies. To aid future researchers, we collate a list of datasets to train and test machine learning classifiers as shown in section IV-B2
5) Highlight limitations and future directions: Finally, in section V, we outline the limitations of existing studies within the ransomware detection domain and present future directions for researchers to consider.

B. ORGANIZATION

The rest of this paper is organised as follows: Section II outlines the motivations and contributions of this paper. Section III looks at current ransomware trends. Section IV reviews the machine learning techniques used to detect ransomware, proposes a taxonomy for ransomware detection

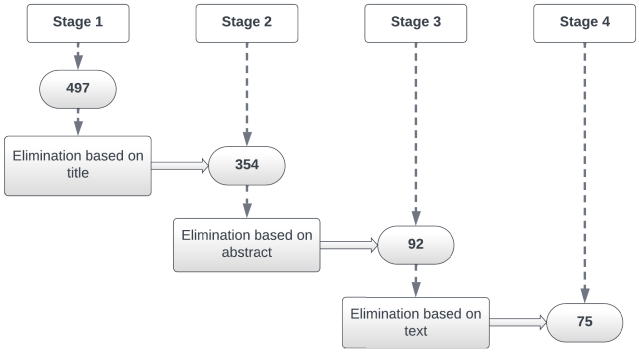

FIGURE 2. The process used to search for and filter relevant literature.

systems, and reviews the detection phase and features, algorithms and designs used throughout the literature to detect ransomware. Finally, in Section V, we discuss the limitations of ransomware detection systems and future directions for research consideration.

II. RESEARCH METHODOLOGY

In this section, we will detail the process undertaken to assess existing studies in ransomware detection. We will explain the methodology employed in this survey for searching and selecting relevant studies and elucidate the inclusion and exclusion criteria utilised to refine the results.

A. SEARCH/DATA SOURCES

Since ransomware constantly evolves, referring to previous studies to determine the latest ransomware trends is impractical. As a result, we consult several grey literature data sources to collate information about the latest tactics, techniques and procedures from ransomware observed in the wild. Specifically, these sources included 1) the European Union Agency for Cyber Security, 2) the Cyber Security and Infrastructure Security Agency, and 3) the Australian Signals Directorate. To collate a collection of ransomware detection studies, we searched several reputable databases to gather appropriate literature. These included: 1) ACM Digital Library; 2) IEEE Explore; 3) Google Scholar; 4) Science Direct; 5) Scopus; 6) Springer; 7) Taylor & Francis; 8) Wiley Online Library.

B. SEARCH KEYWORDS

Search terms were chosen to locate articles on ransomware detection using machine learning. Consequently, we carefully selected the following terms to locate relevant literature within academic archives: 1) 'ransomware'; 2) detection'; 3) 'machine learning'; 3) 'deep learning'. While ransomware is a subtype of malware, we limit our search to 'ransomware' only to align with the scope of the survey.

C. SELECTION CRITERIA

We adhered to the methodology illustrated in Figure 2 for the literature review. Given the specific focus on Windows-based

TABLE 2. A summary of the inclusion and exclusion criteria used to locate studies within the survey.

#	Inclusions	Exclusions
1	The study focuses on ransomware detection using machine learning or deep learning within Windows systems	The study is based on IOT systems, ransomware on mobile platforms, or is not Windows based
2	The article was peer-reviewed and published in a reputable journal or conference paper	The article was not published in a reputable journal article or conference paper
3	The articles are published surveys or research papers	Literature published as news articles or magazines
4	Articles published within the last 5 years	Articles older than 5 years
5	The study was published in English	Studies not written in English

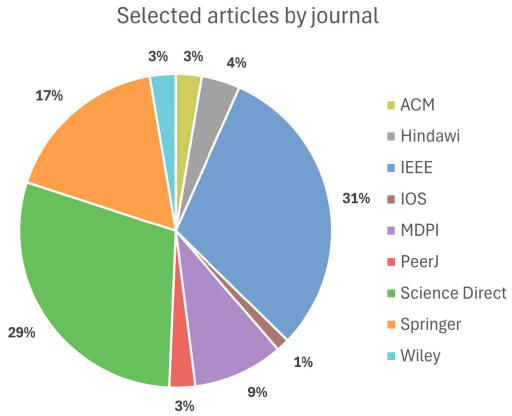

FIGURE 3. Literature included in this survey, segregated by journal.

ransomware, articles about solutions for IoT and mobile platforms, such as Android, were intentionally excluded. Additionally, to streamline the investigation, articles that did not align with the inclusion criteria outlined in Table 2 were excluded. These criteria encompassed factors such as peer-reviewed status, language (English), and direct relevance to ransomware detection. Through this filtration process, the initial corpus of 497 articles was narrowed down to 75.

D. RESULTS

Fig. 3 shows the articles categorised by journal. The largest proportion of articles, 31% (23 articles), were sourced from IEEE, covering a range of publications such as IEEE Access and IEEE Transactions on Computing. Science Direct contributed 29% (22 articles) from various journals, including Computers & Security, Cluster Computing, and Expert Systems with Applications. Springer accounted for 17% (13 articles) from journals like Applied Intelligence and Cybersecurity. The remaining articles, representing 23% (17), originated from ACM, Hindawi, IOS, MDPI, PeerJ, and Wiley. Since 2018, the number of articles related to ransomware detection has grown steadily, as shown in Fig. 4. 28% (21) of articles were published in 2023, which was

TABLE 3. A comparison of surveys related to ransomware detection using machine learning.

Survey	Ransomware trends	Feature types	Algorithms	Detection techniques	Architectural approaches	Real-time detection	Early detection	Available datasets
F. Aldauiji et al. [11]	○	●	●	●	○	○	○	●
A. Alqahtani and F.T. Sheldon [12]	○	◐	◐	●	◐	○	●	○
A. Alraizza and A. Algarni [13]	●	◐	◐	◐	○	◐	◐	○
B. A. S. Al-rimy et al. [14]	●	◐	○	●	○	○	●	○
I. Bello et al. [15]	○	◐	●	◐	○	◐	○	●
E. Berrueta et al. [16]	●	●	◐	●	○	○	○	○
N.M. Chayal et al. [17]	●	●	○	●	○	○	○	○
D.W Fernando et al. [18]	◐	●	●	●	○	○	◐	◐
J. A. Gómez Hernández et al. [19]	●	●	●	◐	○	○	○	◐
A. Kapoor et al. [20]	●	◐	○	●	○	○	◐	○
A. M. Maigida et al. [21]	◐	◐	○	●	○	○	◐	●
T. McIntosh et al. [22]	●	●	○	●	○	○	○	○
R. Moussaileb et al. [23]	○	●	○	●	○	◐	●	○
O. Or-Meir et al. [24]	○	●	○	●	●	○	○	○
H. Oz et al. [25]	●	●	◐	●	○	○	○	●
R. Rani et al. [26]	○	●	◐	●	○	○	○	◐
S. Razaulla et al. [27]	●	●	●	●	○	◐	◐	○
J. Singh and J. Singh [29]	○	●	●	●	○	◐	○	◐
D. Smith et al. [28]	●	◐	●	◐	◐	○	○	○
V. Thangapandia [30]	○	●	●	●	◐	○	◐	◐
U. Urooj et al. [31]	◐	◐	◐	●	○	○	●	●
D. Ucci et al. [32]	◐	●	●	●	○	○	○	●
This work	●	●	●	●	●	●	●	●

○ = No information about topic provided, ◐ = Partial information provided, ● = Comprehensive amount of information provided

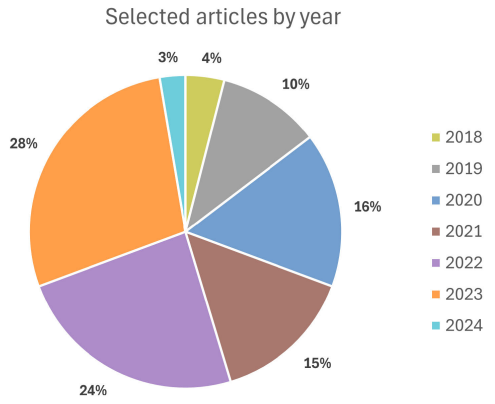

FIGURE 4. Literature included in this survey, segregated by year.

larger than in previous years, indicating steady growth of the domain.

III. UNDERSTANDING RANSOMWARE

This section looks at ransomware in depth. We define ransomware, highlight recent ransomware attacks and discuss ransomware behaviour that has been observed.

A. RANSOMWARE DEFINED

Ransomware has been described throughout the literature as a type of malware designed to extort data and hold it for ransom [33] or as malware that extorts users by locking them out of computer resources and demanding payment to get access back [16]. Overall, ransomware enables cybercriminals to profit from the fear of losing or disclosing sensitive data. Three types of ransomware facilitate this kind of activity, namely:

1) CRYPTO-RANSOMWARE

Most ransomware attacks reported are from crypto-ransomware [2]. Crypto-ransomware (short for cryptographic ransomware) uses cryptography to encrypt the victim's files. Unlike other types of malware, which operate in stealth, ransomware quickly encrypts as many files as possible. Early crypto-ransomware used symmetric encryption keys to encrypt files [34]; however, recent ransomware uses asymmetric encryption to increase the chance of payment. Once the target files are encrypted, the victim is instructed to pay the malicious actor (in cryptocurrencies) for the decryption key to recover the files. Researchers have increased attention to ransomware detection to address the threat. Most of these studies focus on crypto-ransomware since it is the most common ransomware criminal groups use.

2) LOCKER-RANSOMWARE

Otherwise known as Screenlocker ransomware. Locker-ransomware blocks users out of system resources by hijacking system operations such as input devices and applications [14]. Unlike crypto-ransomware, which encrypts user files, locker ransomware often forces the victim to stay on the ransomware screen while leaving the underlying files intact. Removing locker ransomware usually means disabling the malicious process or removing the malicious application. For this reason, it is a less effective means of extortion than crypto-ransomware and, hence, less commonly used by Cybercriminals.

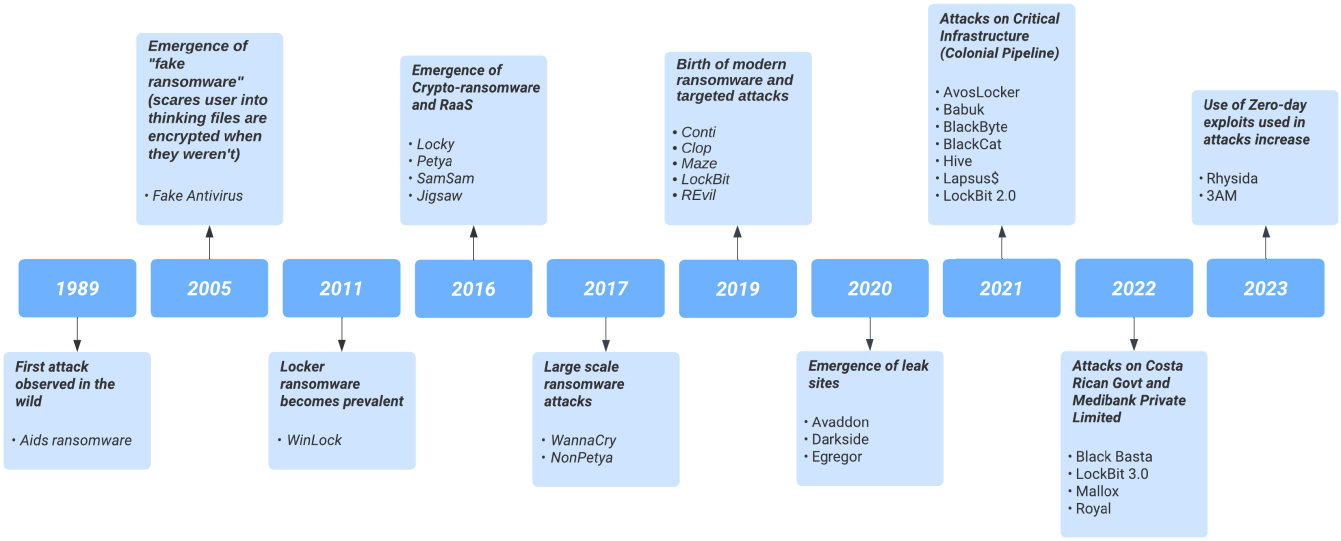

FIGURE 5. A brief timeline of significant ransomware events and families.

B. RANSOMWARE FAMILIES USED IN RECENT ATTACKS

This section outlines several prominent ransomware families recently observed in the wild based on their global impact and provides high-level information about their attacks and techniques.

- *Rhysida (2023)*: Rhysida is a newly identified ransomware strain actively impacting various sectors, including education, healthcare, manufacturing, information technology, and government departments. The actors behind Rhysida specifically target entities with weak authentication controls, utilising remote services like virtual private networks (VPNs) to connect externally. Employing ''living off the land'' techniques for stealth, they utilise RDP for lateral movement and PowerShell. Encryption involves a robust 4096-bit RSA encryption key with a ChaCha20 algorithm, signified by a.rhysida extension on encrypted files. Rhysida employs a ''double extortion'' strategy as explained in section III-C5, threatening to expose sensitive data unless the ransom is paid, aligning with common ransomware practices [35].
- *ALPHV Blackcat (2023)*: Whilst ALPHV Blackcat has been in the wild for a few years, in February 2023 the group announced the release of a new version called ''ALPHV Blackcat Ransomware 2.0 Sphynx update''. ALPHV Blackcat affiliates infiltrate victim networks, employing remote access tools such as AnyDesk, Mega sync, and Splashtop for data exfiltration. They use legitimate tools like Plink and Ngrok for network access, utilising Brute Ratel C4 and Cobalt Strike as beacons for command and control. Evilginx2 facilitates adversary-in-the-middle attacks to acquire multifactor authentication credentials. To evade detection, they leverage allowlisted applications like Metasploit and clear logs on the exchange server. Data is transferred through Mega.nz or Dropbox, and ransomware is activated, accompanied by an embedded ransom note [36]. since their inception, ALPHV Blackcat has siphoned over $300 million in ransom payments.
- *Lockbit3.0 (2022)*: Lockbit stands out as one of the most widespread ransomware strains, functioning under the Ransomware-as-a-Service model. Despite recent iterations, the original Lockbit (version 1.0) surfaced in 2019. Affiliates distributing Lockbit ransomware earn a substantial share, ranging between 60-75% of the illicit profits. Notably, Lockbit has been linked to 16% of all ransomware attacks in the U.S. [37]. The Lockbit group frequently exploits zero-day vulnerabilities for initial access, exemplified by vulnerabilities like Log4Shell and Remote Desktop Services Remote Code Execution. Mimikatz is employed for privilege escalation, while TightVNC facilitates connections with external Command and Control centres, showcasing typical ransomware tactics. Lockbit incorporates advanced evasion techniques, including disabling antivirus software, log removal, and self-deletion. This ransomware strain has been involved in high-profile attacks, such as the Accenture incident, where hackers breached the system and demanded a hefty $50 million ransom [38]. The latest iteration, Lockbit 3.0, introduces bug bounty programs to enhance its software [39].
- *BlackByte (2021)*: A Ransomware-as-a-Service (RaaS) variant that is delivered by phishing or exploiting software vulnerabilities to gain access, such as the unpatched ProxyShell vulnerabilities in Microsoft Exchange Servers [40]. Evades detection by ''living off the land'' or using legitimate software such as certutil to download additional components and the Remote Desktop Protocol (RDP) to establish remote connections. Attacks corporate and critical infrastructure sectors

such as government, financial, food and agriculture facilities.

- *AvosLocker (2021)*: Operates under the RaaS model and targets Linux and virtual machines. Often uses well-known tools such as AnyDesk to establish connections, mimiKatz to exploit authentication protocols, and CobaltStrike to deliver malicious payloads [41]. It encrypts files into the ''.avos'' extension, then directs victims to a ransomware note created in every directory called ''GET_YOUR_FILES_BACK.txt''. AvosLocker is notable for employing affiliates to exploit vulnerabilities in Microsoft Exchange Server and Proxy Shell for ransomware delivery. The malicious actors associated with AvosLocker have a track record of engaging victims directly for ransom negotiations. Moreover, they resort to Distributed Denial of Service (DDoS) attacks to pressure victims into compliance.
- *Hive (2021)*: Gains access through single-factor logins through RDP, Virtual Private Networks (VPNs) or remote network protocols. Once in, the malicious actors exploit multiple Microsoft Exchange vulnerabilities and load malicious backdoor scripts (known as webshells) to allow the attackers to execute malicious PowerShell scripts. Hive disables well-known AV products and uses RDP and Windows Management Instrumentation (WMI) to move laterally. The malicious actors leave a ransom note ''HOW_TO_DECRYPT.txt'' in every directory and publish data on an Onion site if the victim doesn't pay [42]. Responsible for large-scale attacks such as the attack against the Costa Rican Government [43] and the disruption of three hospitals in Ohio and West Virginia, forcing staff to use paper charts [44].
- *Darkside (2020)*: Known for their stealthy techniques in the initial stages of an attack and exploiting compromised contractor accounts and servers to access external servers. Initially establishes Command and Control connections using the Remote Desktop Protocol (RDP) over HTTPS and routed through the TOR network [45]. This ingeniously masks the web traffic to evade detection. Darkside uses well-known tools, such as Mimikatz and psexec, to harvest credentials. Malicious payloads are not deployed until later stages in the attack lifecycle and will cease execution if it is debugged with a VM [46]. It uses process injection to evade detection and, after successful encryption, deletes all tools to remove traces. It gained notoriety for the notable Colonial Pipeline attack, coercing the company to pay a USD 5 million ransom [47] and forcing the price of gas to increase to $3 per gallon. The group behind the attacks is known to launch highly customised attacks and is ruthless in its attacks against hospitals, schools and governments [48]. After the attack, the group behind the malware is thought to have been disbanded (or rebranded) following pressure from the FBI.
- *Clop (2019)*: Inflicting financial damages exceeding $500 million, the group behind Clop recently faced intervention from the Ukrainian police, resulting in the arrest of gang members involved in laundering illicit funds acquired through their attacks [49]. This hacking group is notorious for exploiting zero-day vulnerabilities to breach remote networks, exemplified by the pre-authentication command injection vulnerability on the GoAnywhere MFT platform [50]. Upon establishing connections with external Command and Control servers (C&Cs), Clop downloads widely recognised hacking tools such as Cobalt Strike and Truebot [51]. Clop employs AES, RSA, and RC4 encryption techniques, appending the ''.clop'' extension to encrypted files. Like other ransomware strains, Clop scans running processes to identify security software, disabling or uninstalling them [52]. Notably, Clop checks the keyboard language as a precaution to avoid installation on systems configured with specific languages.
- *REvil (2019)*: One of the most prolific ransomware strains is also known as Sodinokibi [53]. Operating under a Ransomware-as-a-Service model, REvil introduced the double extortion model. Responsible for the infamous Kaseya attack, whereby the malicious actors demanded $70 million. OFAC has designated sanctions against two perpetrators for their role in the Kaseya attack [54]. The pair received over USD 200 million in ransom payments in Bitcoin [2]. Since REvil operates as a RaaS offering, affiliates have used various mediums to distribute the malicious payload. For example, phishing emails compromised RDP sessions and software vulnerabilities seen in the Kaseya attack [55]. Unlike other ransomware samples, REvil uses different encryption algorithms, such as Elliptic-curve Diffie-Hellman, instead of RSA, which is faster by comparison. REvil uses advanced evasion techniques such as process injection, starting in safe mode and disabling security tools. It is known to detect the default system language and terminate itself if the language is found on a list [56].
- *Conti (2019)*: One of the top three ransomware groups that also use the RaaS model and double extortion. Conti also sells access to organisations if ransoms are not paid [57]. Known for attacks against the Costa Rican Government and forcing the country's president to declare a state of emergency [9]. The Conti ransomware encrypts its payload to evade detection from security software and uses DLL injection to load malicious payloads into memory. Once established on a target server, it spreads across the network using Server Message Block (SMB) [58].

C. RANSOMWARE ATTACK STAGES AND TRENDS

Understanding ransomware behaviour involves dissecting the attack lifecycle into various stages, but achieving consensus

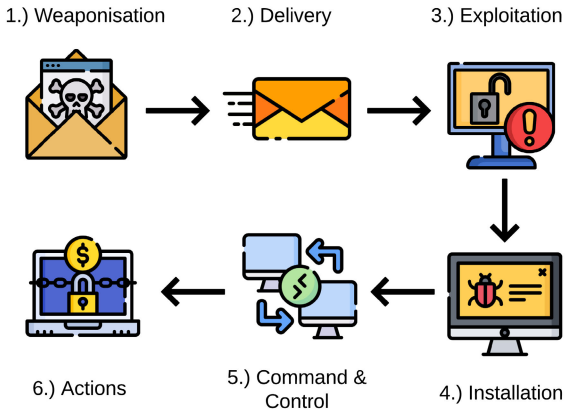

FIGURE 6. Stages of a ransomware attack [60].

on how these stages are defined remains a challenge. Some studies take a holistic approach, categorising ransomware attack stages into reconnaissance, distribution, execution, encryption, and extortion [48]. In contrast, others focus solely on post-infection steps [16], ignoring activities preceding malware execution, such as delivery to the target system. Certain studies adopt established attack frameworks like the Lockheed Cyber-kill-chain [59], outlining phases such as reconnaissance, weaponisation, delivery, exploitation, installation, command and control, and actions on objectives. The reconnaissance phase, challenging to detect and often uncertain regarding the intended target, is omitted in some models [60] and is shown in Fig. 6.

For a successful ransomware attack, malicious files must be delivered to the target machine through phishing, spam, drive-by-downloads, exploits, or social engineering [34]. In the case of crypto-ransomware, after execution, the malware paves the way for file encryption, employing evasion techniques to thwart security software as explained further in section III-C3. Once a foothold is established, the ransomware enters a discovery phase, identifying files to encrypt and gathering crucial information about the target environment [14], [60]. External servers are often contacted for additional modules or file transfers. The ransomware achieves its primary objective by encrypting sensitive data following these steps. The victim receives instructions for file recovery through payment, conveyed through a ransom note or changes to desktop backgrounds. This overview sets the stage for discussing key trends observed in recent ransomware attacks.

1) RANSOMWARE-AS-A-SERVICE

The profitability of ransomware has increased with criminal groups adopting a Ransomware-as-a-Service (RaaS) business model [14]. Similar to Software-as-a-Service, RaaS involves selling or renting ransomware capabilities to affiliates in exchange for a commission [6]. Ransomware groups emulate legitimate software companies, even offering product improvement or bug discovery rewards [39]. Group leaders typically maintain strict control over their software and payment infrastructure, providing affiliates with commissions of up to 60-75% in cryptocurrencies [61]. This model allows ransomware developers to concentrate on their core product while aspects like money laundering or reconnaissance are outsourced to third parties. Despite its advantages, government agencies are taking notice, sanctioning well-known groups like REvil, responsible for extorting over USD 200 million [2].

2) MODULAR NATURE OF RANSOMWARE

Many ransomware detection studies in the literature primarily focused on identifying the actual executable responsible for encrypting files. Consequently, these studies often train and test machine learning models using samples from open repositories like VirusTotal.[5] However, in real-world scenarios, executing the actual ransomware executable, responsible for file encryption, typically occurs later in the ransomware attack lifecycle. In practice, a ransomware attack often involves coordinating various tools and scripts, as seen in notorious ransomware families like Conti. These attacks use proprietary tools like downloaders to fetch additional configurations, CobaltStrike to inject malicious payloads into files, and AnyDesk for establishing connectivity with a Command and Control (C&C) server [58]. The objective is to evade detection, leveraging legitimate tools or those already present on the system (living off the land). Consequently, a common tactic ransomware employs involves connecting to an external Command and Control server (C&C) to download malicious payloads, encryption keys, or exfiltrate data.

3) EVASION TECHNIQUES

Modern ransomware employs sophisticated techniques to avoid detection. In the past, early malware utilised obfuscation methods like polymorphism and code encryption to hide ransomware files. Still, dynamic detection methods discussed in the literature have proven effective against these techniques [62]. In the initial stages of an attack, ransomware commonly uses evasion techniques such as return-oriented programming (ROP) to conceal the malicious file within benign processes [63] or DLL side-loading to inject the malicious payload into a benign process [64]. Additionally, ransomware may utilise fileless malware, such as PowerShell scripts, to disable security software and download the ransomware payload [65]. Because fileless malware operates directly in the system's memory, it often goes undetected by security software [66]. More recently, ransomware has been observed restarting infected systems in safe mode to evade detection by security software that may not load during this mode [67].

4) ENCRYPTION

Arguably, one of the most significant functions of ransomware lies in its advanced encryption methods. In the

[5]https://www.virustotal.com/

early stages, ransomware utilised symmetric encryption to encrypt files rapidly. However, a key issue with symmetric encryption was using the same key for encryption and decryption. If the encryption key was discovered during system forensics, the extortion could be thwarted [68]. This led to the widespread adoption of asymmetric encryption, where different keys are used for encryption and decryption. In this method, the ransomware encrypts files using the public key while holding the private key, which is then sold back to the victim for file decryption. Although effective, asymmetric encryption is notably slower than its symmetric counterpart. Since ransomware aims to encrypt as many files as possible, prolonging the encryption process diminishes the attack's effectiveness. Recently, ransomware has embraced a hybrid approach [69], encrypting files with a symmetric algorithm like AES and employing an asymmetric encryption algorithm such as RSA to encrypt the symmetric key. This encrypted key is then sent to the criminals for sale back to the victim [70]. A recent innovative approach involves partial encryption of files, exemplified by LockBit. LockBit, which claimed the fastest encryption time among ransomware, achieved this by encrypting only the first 4 bits of data [71].

5) DOUBLE EXTORTION

Double extortion is a tactic growing in popularity among ransomware groups. In practice, this involves the attacker exfiltrating the victim's data and threatening to sell, auction, or publish the data on third-party sites. Since ransomware establishes external connections with C&Cs, the ability to exfiltrate data is not hard to include as a feature. This was recently observed in the modus operandi of the Rhysida ransomware family to establish another source of income from the same victim [35]. Since data breaches are often reported at the government level and sometimes incur heavy penalties, the threat of leaking sensitive data to the internet is used as leverage to obtain the requested ransom. However, paying the ransom fee is no guarantee that the data will not be published.

IV. RANSOMWARE DETECTION USING MACHINE LEARNING

In the initial stages, detecting malware relied on file signatures to identify malicious behaviour. Signature-based detection entails comparing file information, such as the MD5 hash, with a database of known malicious file hashes [29]. This method, widely adopted by antivirus software for its high accuracy and ease of implementation, faced challenges in updating databases with new ransomware signatures, as evading detection became simple by altering a few lines of code within the file. In response, researchers delved into behaviour-based detection to identify malicious software.

Unlike signature-based detection, behaviour-based detection seeks to unveil a file's intended malicious actions by analysing it either at rest (static detection) or during execution (dynamic behaviour). Machine learning facilitates automation by creating ML models and comparing the file's behaviour with known benign and malicious files. Although ransomware emerged in 1989 with the AiDS family [72], ransomware detection did not gain prominence until 2016 [22]. By then, machine learning had become a well-explored research area for malware detection. Consequently, machine learning has been a recurrent theme in ransomware detection studies. Early studies heavily relied on traditional ML techniques, while more recent research increasingly adopts deep learning approaches. The taxonomy depicted in Fig. 7 outlines the detection phases, machine learning techniques, and system design approaches utilised in ransomware detection using machine learning. This serves as a summary for the subsequent sections. A comprehensive overview of studies from the literature is presented in Table: 4.

A. RANSOMWARE DETECTION PHASES

In this section, our goal is to comprehend the early detection possibilities within the ransomware attack lifecycle. To achieve this, we align studies with the Cyber-Kill-Chain (CKC) [60] to pinpoint the earliest stage at which ransomware could be detected, as shown in Fig. 6. While ''early detection'' is a frequently used term, it's important to note that most ransomware detection proposals examined in the literature aim to identify ransomware before it encrypts sensitive data. The subsequent sections reveal that researchers have effectively observed ransomware behaviour during its delivery, installation, communication with external servers, and after payments have been made.

1) DETECTION DURING DELIVERY

Numerous studies have proposed solutions for detecting ransomware over the network before it reaches the host. This aligns with the ''delivery phase'' in the Cyber-Kill-Chain, covering ransomware transmission from the source to the intended target. While ''early detection'' is prevalent, it commonly refers to identifying ransomware before it encrypts sensitive data. Yet, most studies asserting early detection typically extract features at the host level, specifically during the installation of ransomware. Given that the ransomware binary is typically deployed in the final stages of an attack, this approach is inherently riskier, with a small window for detection. A different strategy involves detecting ransomware activity before or during transmission before its payload runs on the host. Numerous authors have effectively showcased the success of this approach [73], [74], [75], [76], [77], [78].

Liu and Patras [73] identify ransomware while it seeks new victims on the network via Server Message Block (SMB) protocol requests on port 445. Their method effectively countered WannaCry ransomware, which exploited the EternalBlue vulnerability to access other network systems. Using Bi-ALSTM, the authors detect attack patterns before the network is compromised, achieving an impressive 99.97% detection rate.

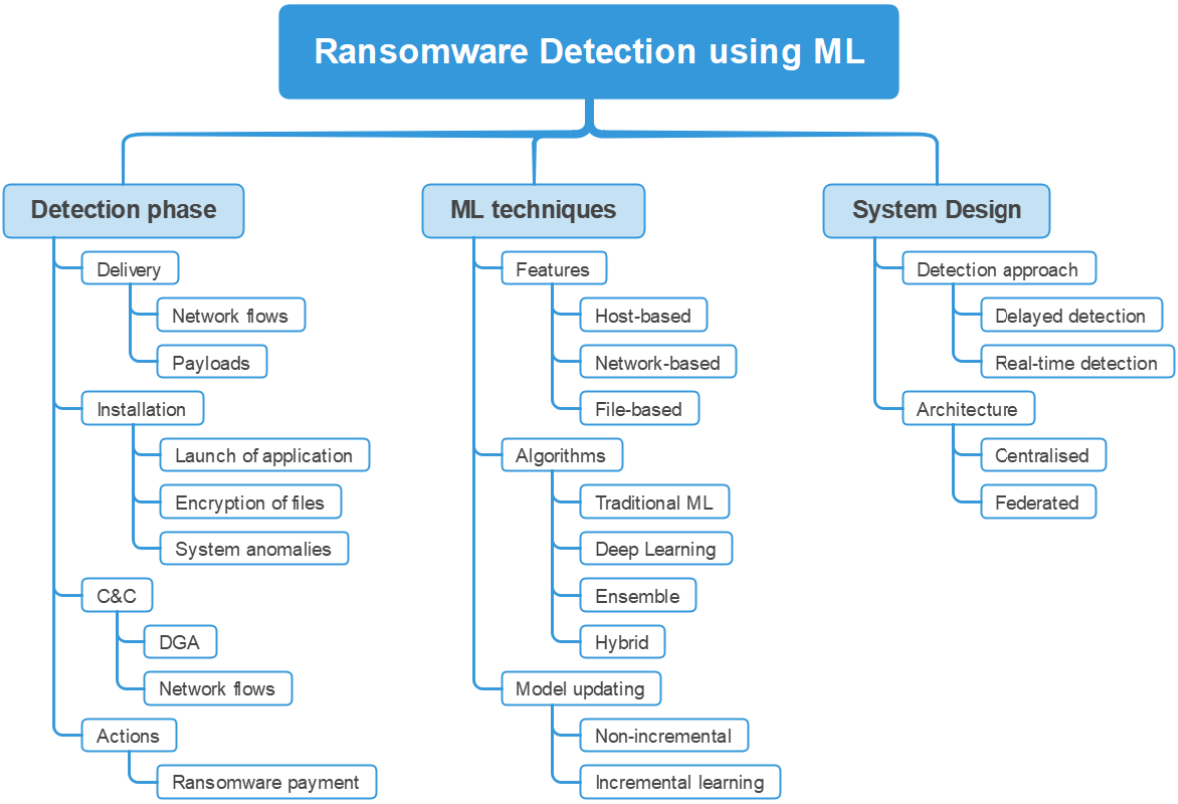

FIGURE 7. Taxonomy of ransomware detection systems proposed throughout the literature.

Berrueta et al. [74] employ a similar strategy, monitoring the communication between clients and file servers using a network probe. This probe captures and analyses file-sharing traffic, including SMB and NFS traffic. In their study, the authors extract features from network traffic, such as file reads, writes, deletes, and rename actions. These features are utilised to train and test various algorithms, including decision trees, tree ensembles, and neural networks, resulting in the detection of ransomware activity with an impressive accuracy of 99.9%. The advantage of this approach lies in the fact that the probe analysing the traffic is not directly exposed to the ransomware, as the tool operates off-path. However, this approach is vulnerable to file-less ransomware.

Maimó et al. [75] have developed a ransomware detection system specifically designed for integrated clinical environments (ICE). Given that certain clinical data, much like some ransomware traffic, travels in an encrypted form across the network, examining individual network packets becomes challenging. To overcome this, the authors monitor network flows instead of scrutinising each packet separately. Their study uses a sliding window technique to observe network traffic for anomalies. The features captured include TCP/UDP network features such as source IP, destination IP, and destination port. The study utilises the One-Class Support Vector Machine (OC-SVM) anomaly detection algorithm to pinpoint unusual traffic, and the Random Forest algorithm is applied to categorise the identified traffic patterns as malicious or benign.

An alternative to detecting ransomware behaviour during transit is post-delivery detection [79], for example, whilst the file has reached its destination but has not been activated. This often involves extracting the static features of the file, such as printable strings and opcodes, and classifying these files before installation. However, ransomware can easily evade this approach using obfuscation techniques such as polymorphism and encryption [22].

While detecting ransomware during its delivery has proven effective, it comes with various challenges, including: (1) Obfuscation, encrypted payloads, or encrypted tunnels often hinder detection; (2) Finding features without needing the host to be infected is challenging; (3) Packet inspection is an expensive and difficult task.

2) DETECTION WHILE RANSOMWARE IS INSTALLING OR RUNNING

The prevalent method for ransomware detection involves monitoring its activities while it is running or post-execution, aligning with the installation phase of the Cyber-Kill-Chain. This period spans from the execution of the malicious binary to the encryption of sensitive data and the display of the ransomware note. In the existing literature, instances of detecting

ransomware behaviour during this phase are illuminated by: (1) the initiation or execution of an application or process; (2) observed modifications to files, or; (3) the identification of system-wide anomalies, such as changes in CPU activity. Consequently, most studies derive features from the host, incorporating data such as system calls (syscalls) or Windows Application Programming Interfaces (APIs). To categorise malicious files, studies employ techniques like monitoring the execution cycles of ransomware or transferring files to secure virtual machines (VM) for detonation and subsequent analysis.

Syscalls prove effective in capturing suspicious behaviour immediately after the launch of a malicious file. Ransomware necessitates interaction with the executive layer of the operating system (OS) to effect system-wide changes, such as deleting shadow files. A common approach involves extracting syscalls at set time intervals and subsequently modelling the behaviour based on this data [80]. While studies have demonstrated accurate results in modelling ransomware behaviour using syscalls, it's important to note that extracting syscalls is more intricate than pulling API calls and demands a kernel driver for real-time ransomware detection. Despite this complexity, syscalls offer practical features for ransomware detection because: (1) standard API calls are resource-intensive; (2) API calls provide limited telemetry at the user level, and; (3) syscalls offer visibility into rootkits and low-level malware activities, such as the deletion of backup files [81].

An alternative strategy involves deploying decoy files to identify the presence of ransomware on a host. Shaukat and Ribeiro [82] employed this approach by monitoring for changes to decoy files. This method proves accurate for detection, as there is circumstantial evidence of ransomware following tampering with a file. However, detection typically occurs only after files are encrypted, providing a limited time window for ransomware removal. Another approach involves monitoring system logs for malicious behaviour. In contrast to methods that require a trigger for detection, such as the launch of an application, holistic system tracking can identify malware even when advanced evasion techniques, such as process injection or hiding behind multiple processes, are employed. Roy and Chen [83] monitor for anomalies using the Windows Logging Service (WLS) to extract Event ID sequences. Once extracted, these sequences are sent to a centralised server on the network, which employs BiLSTM-CLF for classification. Detecting ransomware during the installation phase of the attack lifecycle has proven to be effective, given the diverse range of features available for detecting malicious behaviour.

3) DETECTION WHILE RANSOMWARE IS COMMUNICATING WITH AN EXTERNAL C&C

An additional method for ransomware detection involves monitoring communication with an external server, commonly called a Command and Control centre (C&C). Once persistence is established on a target machine, ransomware frequently communicates with an external server to facilitate tasks such as exchanging private encryption keys, downloading supplementary modules, or exfiltrating sensitive data. To circumvent network-level blocking of external domain names, ransomware often incorporates domain generation algorithms (DGA) to evade detection. This incorporation enables ransomware to communicate with an external server without disclosing hardcoded IP addresses, complicating forensic efforts.

Almashhadani et al. [84] applied machine learning to identify suspicious communications within a network and detect domain names that resemble those generated by domain generation algorithms (DGAs). Utilising 16 semantic features, including metrics like the number of vowels and entropy calculation for suspicious domain names, the authors employed KNN for classification, achieving a notable detection accuracy of 98%. Despite this success, the study acknowledges potential evasion techniques, such as using shortened URLs or alterations to reduce the entropy of malicious domain names. Most ransomware communicates with an external server; however, many studies assume such communication will occur. Consequently, if ransomware opts not to communicate with an external Command and Control (C&C), detecting it during the C&C phase becomes futile. Encryption poses an additional challenge during this phase, as many malware samples establish connections through encrypted tunnels or onion networks like TOR, making detection more challenging.

4) DETECTION AFTER THE RANSOM HAS BEEN PAID

An emerging area of research involves detecting ransomware activity even after the ransom has been paid. By leveraging the transparent nature of cryptocurrency flows, some studies utilise machine learning to identify transactions facilitating ransom payments to malicious actors. While this method doesn't prevent ransomware from executing or encrypting files, it provides insight into the extent of infection caused by a particular ransomware strain. This approach proves beneficial in threat intelligence scenarios, helping to gauge any uptick in ransomware transactions.

Al-Haija and Alsulami [85] employ the Bitcoin network for ransomware activity detection. Their study utilises a dataset comprising around 3 million transactions, segmented into 41,413 ransomware transactions and 2.9 million benign transactions. They use 10 features, including Bitcoin address, year, count, income, etc., to train and test supervised machine learning models like shallow neural networks (SNN) and optimisable decision trees (ODT). The authors achieve a remarkable 99.9% accuracy in classifying transactions as benign or malicious and a 99.4% accuracy rate in classifying ransomware into their respective families.

Alsaif et al. [86] adopts a similar approach, extracting 18 payment-related features such as Bitcoin address, transaction day, transaction amount, and timestamp from the

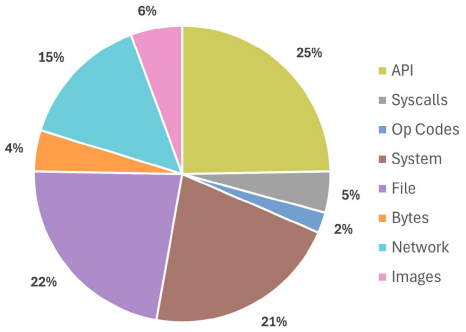

FIGURE 8. Features used in surveyed studies.

same dataset. The author then employs supervised machine learning models like Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) for classifying ransomware transactions, achieving an impressive 99.08% accuracy.

Overall, leveraging the Bitcoin network proves to be successful in detecting ransomware transactions. This method's utility extends to threat detection, and if integrated into Bitcoin miners, it can add value by flagging suspicious transactions early on.

B. MACHINE LEARNING TECHNIQUES

In this section, we examine the various machine learning (ML) techniques employed in ransomware detection as documented in the literature. Our objective is to introduce a taxonomy for categorising ransomware detection systems based on the detection phase, ML techniques utilised, and the design approach employed in their construction. Also, we provide an overview of datasets frequently used in the literature to help guide future research efforts.

1) FEATURES USED TO DETECT RANSOMWARE

Cybersecurity researchers employ both static and dynamic features for ransomware detection. Static features, extracted from executable files at rest, include printable strings, Opcodes, and function calls. While effective in detecting malicious activity, static-based detection can be evaded through obfuscation. Dynamic features acquired at runtime offer a robust detection method. Unlike static features, dynamic ones are harder to obfuscate as their extraction occurs after the malware reveals the executable's true nature. Static features may be ineffective in specific scenarios, such as fileless malware, which doesn't save a file on the hard disk for static feature extraction. Researchers typically utilise combinations of static and dynamic features from the host, network, and file system for comprehensive ransomware detection. Fig. 8 illustrates the distribution of features utilized across the literature.

a: HOST FEATURES

The prevalent approach for detecting ransomware activity is at the host level, where the malware executes. Ransomware detection at runtime involves various methods, including API calls [90], [92], [94], [96], [97], [101], system calls (syscalls) [119], [122], system features like running processes, DLL, and registry entries [90], OpCodes [124], bytes [95], [131], and hardware features [116], [125]. Among these, Windows Application Programming Interfaces (API) calls are extensively used for ransomware detection [90], [92], [94], [96], [97], [101]. The Windows API facilitates interactions between programs and the operating system. With Intel x86 CPUs employing four protection rings, lower-level protection rings have elevated privileges [135]. In the Windows ecosystem, applications in user space operate with limited system privileges, while kernel-level access is sought for system resources. To request these resources, applications use API calls through Kernel32.dll, channelled to the kernel via ntdll.dll, as depicted in Fig. 9. This chokepoint is an effective location to capture API calls commonly associated with malicious intent. As file encryption occurs at the system level, API calls provide a means to detect ransomware before the encryption process begins.

API calls have been employed in various manners in the literature to identify ransomware. Alqahtani and Sheldon [119] focus on directly detecting ransomware activity within cryptographic API calls. Cryptographic API calls play a crucial role in ransomware attacks as they are directly associated with the encryption process employed by ransomware to encrypt user files and data. By monitoring and analysing these API calls, the early detection model can recognise the initial stages of ransomware activity, enabling pre-emptive measures before encryption occurs. Using Deep Belief Networks (DBN), the authors achieve a ransomware detection accuracy of 94.6%. However, this approach assumes that the pre-encryption phase of a ransomware attack is constant and can be defined by static attributes such as time or specific API calls. The limitation is that malicious attackers may use obfuscation techniques to evade detection.

Karbab et al. [115] adopt a method of classifying API sequences to identify ransomware. In their research, the authors execute ransomware in sandboxes, label each report, convert the reports into sequences of words, and then employ the Long Short-Term Memory (LSTM) classifier for classification. The authors observe an F1-score of 93.66% in their production environment and a low false positive rate of 0.99%. However, other studies show that ransomware authors can elude detection by employing code obfuscation, anti-virtualisation or API spoofing. API spoofing becomes a notable concern, especially when the monitored API calls originate from the user level of the operating system, as applications can easily trigger dummy API calls to hide malicious intentions.

Detecting ransomware through system calls (syscalls) extracted at the executive level of the operating system provides an alternative approach. When invoked by the API, the ntdll.dll file in the user space calls the Ntoskrnl.exe file, which is then directed to the system drivers. Researchers have exploited this bottleneck to identify malicious calls

TABLE 4. Summary of Ransomware Detection Systems proposed throughout the literature.

Study	Algorithms	Datasets	Real time?	Features								Performance				
				API	Syscalls	Op Codes	System features	File features	Bytes	Network features	Images	Accuracy	Precision	Recall	F1	False positives
Roy and Chen [83]	BiLSTM, CRF	17 Ransomware samples	●				●			●		99.67%				
Hsu et al. [87]	SVM	4 Ransomware samples, 22 benign files	○					●				92%	94%	84%	88%	
Poudyal and Dasgupta [88]	LR, SVM, RF, J48 AdaBoost (RF/J48), Neural Network	2600 Malware samples, 550 Ransomware samples from VirusTotal. 540 Goodware	○				●					99.72%				0.1%
Aurangzeb el al. [89]	DT, RF, Gradient Boosting, Extreme Gradient Boosting	160 Malware samples from VirusShare	○				●							94%		
Molina et al. [90]	Naive Bayes, KNN, RF, ANN, LSTM, BiLSTM	90,364 Malware samples from VirusTotal, 39,136 from VirusShare (distilled into 19,499 Ransomware samples from 2010-2019 from 21 families	○	●			●					94.92%	95.16%	94.44%	94.61%	
Almashhadani et al. [84]	DT, Ensemble tree, Naive Bayes, SVM, KNN	85,000 malicious Domain Generation Algorithms (DGA) from 20 Ransomware families. 85,000 Goodware	◐							●		94.52%				4%
Ahmed et al. [91]	DT, KNN, LR, RF, SVM	1,354 Ransomware samples from 14 families (from VirusShare and VirusTotal) 1,358 Goodware	○	●								97.4%				1.6%
Sharmeen et al. [92]	Deep Learning, SVM, RF, Multi-class classifier	1,232 Ransomware samples from 14 families from VirusShare and VirusTotal. 1,308 Goodware samples from Windows 7	○	●								95.96%				
Khan et al. [93]	MOGWO (Deep learning Swarm Intelligence based algorithm), Naive Bayes, AdaBoost, Decision Stump	582 Ransomware samples, 942 Goodware samples	○									87.91%				12%
Kok et al. [94]	RF	904 Ransomware from VirusShare and 942 Goodware samples	◐	●								100%				0%
Khammas [95]	RF	840 Ransomware from VirusTotal (Cerber, TeslaCrypt and Locky), 840 Goodware	○						●			97.74%				0.6%
Kok et al. [96]	RF	1,846 Ransomware and Goodware from VirusTotal and theZoo	◐	●								99%				
Bae et al. [97]	RF, Naive Bayes, LG, KNN, SVM, Stochastic Gradient Descent (SGD)	1,000 Ransomware, 900 Malware, 300 Goodware samples	○	●								98.65%	98.25%	98.94%	98.54%	
Keong et al. [98]	Extra tree classifier, RF, Gradient Boosting, KNN, LR, Guassian Process, SVM	1,000 Ransomware samples divided into 18 families	◐	●				●				96.53%	96.23%	96.44%	96.25%	
Homayoun et al. [80]	LSTM, CNN, MLP	220 Locky, 220 Cerber, 220 TeslaCrypt, 99 CryptoWall, 28 TorrentLocker, 77 Sage and 220 Goodware from 2016-2017	●		●							97.2%				2.7%
Almashhadani et al. [99]	RF, LibSVM, Bayes Net, RT	Malware Capture Facility Project (MCFP) dataset with Locky samples from VirusShare	◐							●		98.72%				2.1%
Lee et al. [100]	KNN, Linear Model, DT, DT Ensemble, Kernel Trick, Neural Network	1,200 files encrypted by Ransomware	●					●				100%				
Kok et al. [101]	PEDA, RF, Naive Bayes, Ensemble (RF and Naive Bayes)	582 Ransomware and 942 Goodware	◐	●								99.3%				1.56%
Shaukat and Ribeiro [82]	LR, SVM, ANN, RF, Gradient Tree Boosting	574 Ransomware from VirusShare from 12 families, 442 Goodware	◐	●			●					98.25%				0.56%
Cohen and Nissim [102]	Naive Bayes, Bayes Net, J48, RF, LR, LogiBoost, SMO, Bagging, AdaBoost	Ransomware including Cerber, TeslaCrypt, Vipasana, Chimera and HiddenTear	○				●					92.2%				5.2%
Nissim et al. [103]	DT, RF, Naive Bayes, Bayesian Network, SVM	Ransomware including Cerber, TeslaCrypt, Vipasana, Chimera and HiddenTear	○		●							97.9%				0%
Abbasi et al. [104]	Regularized Logistic Regression (RLR), RF, DT, SVM, kNN	Resilient Information Systems Security (RISS) dataset with 582 ransomware and 942 goodware	○	●			●	●				94.33%				
Masum et al. [105]	DT, RF, NB, LR, NN	138,047 Ransomware samples from VirusShare	○					●				99%	99%	97%	97%	
Chaganti et al. [106]	CNN, CNN-LSTM, DNN	1,500 malware samples from Virusshare.com and 875 benign files from portableapps.com	○	●				●					96%	96%	96%	
Li et al. [107]	XGBoost	CCF BDCI-21 (5,841 malware samples) Microsoft BIG-15 (10,868 samples)	○			●		●				99.2%				
Ba'abbad and Batarfi [108]	Hoeffding Tree Classifier (HTC)	Resilient Information Systems Security (RISS) dataset containing 582 ransomware and 942 goodware	○	●			●	●				99.4%				
Woralert et al. [109]	LSTM	Various Ransomware samples from MalwareBazaar	○				●									
Deng et al. [110]	Deep reinforcement learning (DRL) based on Double Deep Q Network (DDQN)	35,367 samples from VirusShare 27,118 benign samples from Windows 688 samples from [111]	◐					●				97.9%	97.4%	97.9%	97.7%	
Thummapudi et al. [112]	RF, DT, SVM, kNN, XGBoost, DNN, LSTM	100 samples from VirusShare	○				●					92.3%	99.5%	97%	95.6%	3%
Gulmez et al. [113]	CNN, XAI (LIME and SHAP)	5 datasets: 6,263 from VirusShare, 7,703 from Sorel-20M, 668 from ISOT, 6,263 malware from VX Heaven and 14,797 benign samples from informer.com	○	●			●					98.2%				4.7%
Fernando and Komninos [114]	RF, LR, SVM, J48, Gradient Boosting Trees, BN, MLP, SGD	720 ransomware samples, 2000 benign	○	●								94.3%	94.3%			0.8%
Karbab et al. [115]	CNN, LSTM, MLP	45k Benign, 38k Ransomware	○	●				●		●					93.66%	1.93%
Anand et al. [116]	Boruta algorithm / Random Forest	183 ransomware samples from Malware Bazaar	○				●					98.68%				
Gazzan and Sheldon [117]	Generative Adversarial Network (GANs), CNNs, LSTM	8,152 from VirusShare and 1000 benign samples from informer.com	○	●								98%		96%		0.14%
Li et al. [118]	LightGBM, CNN, RNN (LSTM, GRU), Bi directional RNN (Bi-LSTM, Bi-GRU)	BGP update messages collected from Jan-21-21 until Jan-31-21 (11 days) from RIPE remote collector and route views collector TELXATL	○							●		84.27%	84.23%		86.90%	
Alqahtani and Sheldon [119]	SVM, LR, Deep Belief Networks (DBN), CNN, MLP	39,378 ransomware files from VirusShare	○	●	●							94.60%	94.20%	97.40%	95.90%	
Singh et al. [120]	CNN, Pre-trained transformer algorithms	Cloud encrytped dataset	○				●	●					99.50%	98.50%	97.64%	
Alsaif et al. [86]	LR, RF, XGBoost	41,413 ransomware transactions from the UCI Machine Learning Repository	○									99.08%	99.86%	99.16%	99.5%	

○ = Not observed, ◐ = Partially observed, ● = Observed

System features = Features derived from the host such as registry keys, mutex's, CPU, etc. File features = Features derived from files such as file hashes, metadata, etc. Network features = Features derived from the network such as IP addresses

TABLE 4. *(Continued.)* **Summary of Ransomware Detection Systems proposed throughout the literature.**

Study	Algorithms	Datasets	Real time?	Features								Performance				
				API	Syscalls	Op Codes	System features	File features	Bytes	Network features	Images	Accuracy	Precision	Recall	F1	False positives
Ayub et al. [121]	Ensemble: DT, RF, AdaBoost, GB, SVM	215 Cyrpto ransomware from Sophos ReversingLabs (SOREL-20M), 3, 014 benign applications	○	●			●	●				91.75%	91.99%	90.47%	91.05%	
Chaithanya and Brahmananda [122]	AdaBoost, XGBoost, LightGBM, SVM, KNN, NB	VirusTotal dataset with ransomware from 21 different families	○		●		●	●		●		91%				
Ganfure et al. [123]	Affinity Propogation (AP)	1,106 ransomware samples from 20 different families from VirusShare. 11,311 benign files	●				●									0.20%
Ciaramella et al. [124]	CNN, VGG-16	15,000 malware samples with 5,000 ransomware samples from VirusShare	○			●					●	96.90%	97%	96.90%		
Singh et al. [76]	DT, SVM, RF, KNN, SGD, ANN	70 ransomware samples from 31 families	○							●		99.83%	99.82%	99.83%	99.83%	
Ganfure et al. [125]	CNN, OC-SVM, RATAFIA, EGB	515 ransomware samples from 21 families from VirusShare	○					●			●		98.20%	98.60%		
Wazid et al. [126]	RF, LR, DT, KNN	BitcoinHeist Ransomware Address Dataset. Containing 2,916,697 benign transations and 41,413 ransomware related transactions	○									98.98%			99.90%	
Prachi and Kumar [127]	RF, SVM, kNN, LR, NB	50 ransomware samples from 10 families	○	●			●	●		●		99%	99.20%	98.90%	98.80%	0
Zahoora et al. [128]	Deep Contractive Autoencoder (CAE), SVM, RF, LR	RISS dataset containing 582 ransomware and 942 benign samples	○	●			●	●				93%		99%	93%	
Aurangzeb et al. [129]	SVM, RF, KNN, XGBoost, NN	582 ransomware (11 families) and 942 benign samples from VirusShare	○	●			●	●				98.00%	98%	94%	94%	
Berrueta et al. [74]	Decision Trees (DT), Tree Ensembles (TE), Neural Networks (NN)	70 ransomware programs from Hybrid Analysis and Malware Traffic Analysis (giving 150 traffic traces in total)	◐							●		99%	99.7%	100%	99.87%	
Zahoora et al. [130]	Deep Contractive Autoencoder (DCAE) with Zero-shot learning, RF, GNB, SVM, LR	582 ransomware samples and 942 goodware samples from VirusShare	○	●			●	●				92.80%		95%		13%
Almashhadani et al. [77]	DT, KNN, SVM, Discriminant analysis, bagged tree	Session-based: 4,128 ransomware, 4,128 benign time-based: 445 ransomware, 445 benign from Malware Capture Facility Project (MCFP)	◐							●		99.88%	99.76%	100%	99.88%	0.024%
Kim et al. [131]	SVM, Neural network	211,807 ransomware files (8lock8, Powerware, Jigsaw, Maktub, CryptoJoker) from	○						●						0.994	
Du et al. [132]	KNN, RF, DBSCAN	9,458 ransomware from 25 families	○						●		●	99%	98%	99%	99%	
Zhang et al. [78]	DCGAN (Deep convolutional GAN), TGAN (Transfer GAN)	CICIDS2017, KDD99, SWaT, WADI	○							●	●		98.10%	9.28%	98.70%	
Rhode et al. [133]	RNN, MLP, SVM, NB, DT, GBDT, RF, AdaBoost	3,604 benign samples and 2,792 malware from VirusTotal and Windows 7	●					●		●					81.5%	14%
Wong et al. [134]	Transfer learning: ECOC-SVM	Malimg, MaleVis, Virus-MNIST, Dumpware10	○						●		●	98.87%				
Maimo et al. [75]	Anomaly detection: OC-SVM for anomaly detection, Naïve Bayes (NB) for classification	Clear traffic, Ransomware samples include WannaCry, Petya, BadRabbit, PowerGhost (50,537 ransomware samples), 100,000 clean samples	●							●		99.99%	92.32%	99.97%	95.96%	4.6%

○ = Not observed, ◐ = Partially observed, ● = Observed

System features = Features derived from the host such as registry keys, mutex's, CPU, etc. File features = Features derived from files such as file hashes, metadata, etc. Network features = Features derived from the network, such as IP addresses

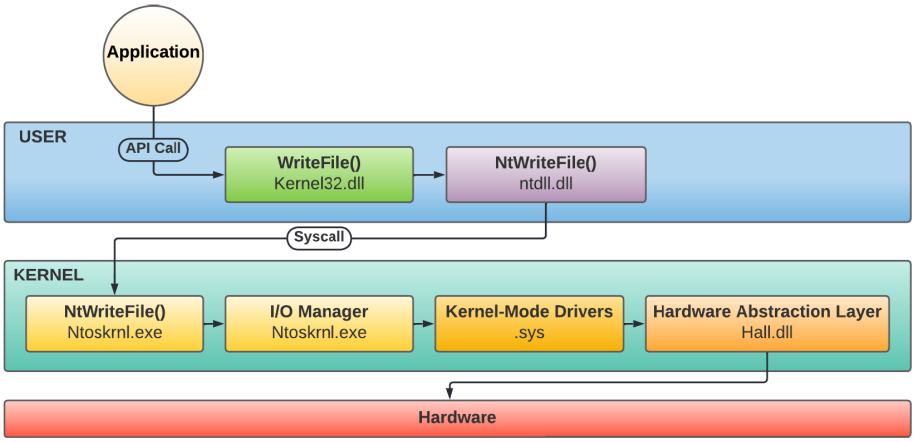

FIGURE 9. **Flow of events through the Windows System Architecture [136].**

to the kernel. In a study by Nissim et al. [103], volatile memory dumps were utilised to analyse syscall behaviour. In their study, the authors detect ransomware activity with an accuracy of 97.5% using the random forest algorithm. Unlike API calls, syscalls exhibit greater resistance to obfuscation, given that they are disclosed at runtime and cannot be directly invoked.

Cohen and Nissim [102] also employed volatile memory dumps to model host behaviour but focused on 23 operating system features such as DLLs, processes, mutexes, handles, services, and threads, instead of relying solely on APIs or syscalls. Classification of memory dumps using random forest resulted in a notable 92.2% accuracy rate. While volatile memory dumps effectively detect ransomware, the drawback lies in the substantial storage needed to store dump files extracted regularly. Consequently, malicious activity may be identified after ransomware has established a foothold. Moreover, recent advancements in evasion

techniques, such as process injection, multiple collaborative processes, and return-oriented programming (ROP), enable malware to conceal itself behind benign processes. Despite the challenges associated with using host-based features for ransomware detection, it remains the most common approach due to its ease of implementation and the abundance of available data for classification.

b: NETWORK FEATURES

Detecting ransomware over the network involves monitoring data in transit, aiming for early identification before infection, which is considered low-risk and less invasive. However, in practice, the encryption of malicious payloads by cybercriminals poses a detection challenge. Consequently, cybersecurity researchers have proposed various features for extraction to enhance the ability to detect ransomware activity over the network. Almashhadani et al. [99] introduced an early ransomware detection system leveraging network features. Their study identifies ransomware before it can compromise the host, utilising 18 packet and flow-based features, including TCP, HTTP, DNS, and NBNS traffic from PCAP files. The authors train and test their machine learning model, achieving a noteworthy 98.72% detection accuracy with the random forest algorithm.

Almashhadani et al. [84] adopt a distinct approach, employing machine learning to identify malicious URL calls from within the network. Their study capitalises on the necessity for ransomware to establish communication with external Command and Control (C&C) servers for tasks like downloading additional modules or exfiltrating data. To prolong the infection period, many malware variants utilise domain generation algorithms (DGA) to avoid detection, preventing network administrators from effectively blocking hardcoded URLs and simplifying forensic efforts. The authors extract pertinent features by analysing 16 characteristics from domain name strings in incoming DNS request packets. These characteristics encompass metrics such as the count of vowels in the URL and entropy. Utilising the K-Nearest Neighbours (KNN) algorithm, they attain an accuracy detection rate of 94.52%. However, it is crucial to recognise that the encryption of packets may hinder the effectiveness of this approach.

Li et al. [118] identify anomalies in the Border Gateway Protocol (BGP) logs obtained during the WestRock ransomware attack, signalling the presence of ransomware activity. BGP is a path-vector routing protocol crucial for determining optimal routes for data packets between different networks. Its significance in internet operations lies in enabling autonomous systems to communicate and make routing decisions based on factors like network policies and path preferences. The authors gathered publicly available BGP records from major internet exchange points globally and extracted 37 features, including metrics like average edit distance and duplicate announcements. Following the training of a LightGBM algorithm, ransomware activity is detected with an accuracy rate of 84.27%.

The mentioned studies employ network-based features to identify ransomware activity when the ransomware communicates with other nodes. However, features like DNS records and BGP records mainly reveal ransomware behaviour after an attack has commenced, and their primary utility lies in curtailing the ransomware's further spread. An alternative approach to detecting ransomware is intercepting and executing payloads before they reach the host, a method embraced by Keong Ng et al. [98]. In their study, the authors intercept executable files through the Suricata inline Intrusion Prevention System (IPS) at the network's gateway entrance. Upon detecting executable files, 54 static features, including header size and checksum, are extracted and used for classification. Suspicious files are directed to a secure virtual machine for detonation, enabling the classification of their dynamic features. However, a limitation of this approach is its inability to detect encrypted data transmitted through tunnels or ransomware delivered via drive-by downloads.

While network-based features show promise in early ransomware detection, researchers encounter challenges posed by evasion techniques such as encryption and tunnelling and ensuring the availability of features to substantiate ransomware activity without its execution on the host.

c: FILE-SYSTEM FEATURES

Another widely used method for ransomware detection involves analysing static files for suspicious activity. Researchers have employed various techniques to identify ransomware activity within files, focusing on the file content or metadata. The content-based approach involves extracting features from the binary file, such as its code, headers, or API calls. Alternatively, the metadata-based method involves examining attributes like file size, entropy, file hash, etc.

Lee et al. [100] employed machine learning to identify changes in file entropy within the system. In their study, the authors gauge file entropy on the network and apply various algorithms such as SVM, KNN, logistic regression, Decision Tree, random forest, and gradient boosting. Despite achieving 100% detection accuracy using this method, it's worth noting that the study does not include legitimately encrypted files in their dataset, raising questions about the real-world applicability of their perfect result.

Similarly, Hsu et al. [87] adopt a comparable approach, utilising entropy to identify files that have already been encrypted by ransomware. They train SVM using 17 file-based features, including the compression ratio of the file and report a detection accuracy of 92% in their experiments. However, the study notes a high false positive rate attributed to the challenge of distinguishing legitimately encrypted files. One significant challenge associated with this approach is that encrypted files, whether malicious or benign, typically exhibit higher entropy [137]. Consequently, it is inaccurate to assume that files with higher entropy are inherently malicious.

TABLE 5. A selection of publicly available datasets used throughout the literature for Ransomware detection research.

Year	Dataset name	Description	Location
2015	Stratosphere IPS Feeds [138]	Real-time malware traffic captures consisting of malware captures, normal activity and mixed captures	[138]
2015	The UNSW-NB15 Dataset [139]	100GB of raw traffic (TCP dumps) generated in the Cyber Range Lab, UNSW Canberra. Consists of real and synthetic attack behaviours	[140]
2015	Microsoft Malware Classification Challenge (BIG 2015) [141]	Roughly half a Terrabyte of data from 9 known Malware families collated from Microsofts real-time detection	[142]
2016	Resilient Information Systems Security (RISS) - Ransomware Dataset [143]	Dynamic analysis of 582 Ransomware samples and 942 benign samples.	[144]
2018	MALREC [145]	66,301 full system recordings (system dumps) from 2014 to 2016, consisting of mixed Malware including Ransomware	[146]
2018	Malware Benchmark for Research Dataset (EMBER) [147]	1 Million PE static files scanned in or before 2018	[148]
2018	Dynamic Malware Analysis kernel and user-level calls [149]	Kernel and User level calls extracted from Cuckoo sandbox of 1000 malware and 1000 clean samples.	[149]
2020	Data breaches and Ransomware attacks from 2004 to 2020 (The University of Queensland) [150]	A public dataset consisting of data breaches and Ransomware attacks from 2004 to 2020	[150]
2020	ISOT Ransomware Detection Dataset [151]	420GB of Ransomware and benign execution traces. Consisting of 669 ransomware samples and 103 benign samples run on a Cuckoo sandbox with Windows 7 (64bit)	[152]
2020	SOREL-20M: A large scale benchmark dataset for malicious PE detection [153]	A dataset of 10 million disarmed malware files and 20 million extracted features. The dataset contains around 1 million ransomware samples	[154]
2020	BitcoinHeist Ransomware Address Dataset [155]	Bitcoin transactions from January 2009 until December 2018. Contains 2.9 transactions with nearly 41,413 malicious transactions.	[155]
2021	BODMAS [156]	Static features extracted from 57,293 malware samples (821 ransomware samples included) and 77,142 benign samples that were collected from August 2019 until September 2020	[157]
2021	Ransomware and user samples for training and validating ML models [158]	File-sharing traffic analysis of more than 70 ransomware binaries from 26 families and more than 2,500 hours of benign traffic	[158]
2022	NapierOne [159]	Nearly 500k files encrypted from various ransomware families such as Notpetya, Lockbit, Maze, Phobos, Netwalker, Dharma and Ryuk	[160]
2022	RanSAP [161]	Storage patterns for 7 (TeslaCrypt, Cerber, WannaCry, GandCrab, Ryuk, Lockbit and Darkside) ransomware samples and 5 benign samples	[162]
2022	Open repository for the evaluation of Ransomware Detection Tools [163]	PCAP logs extracted from more than 90 ransomware files (since 2015) from different ransomware families. Logs include DNS/TCP connections and Input/Output (I/O) operations	[163]

Wong et al. [134] identify dormant ransomware files using a vision-based approach. In their study, binary files are transformed into images, and deep learning is employed to extract features. Finally, an ensemble configuration of Support Vector Machines (SVM) using Optimal Error Correction Output Coding (ECOC) is employed for classification, resulting in a detection accuracy of 98.87%.

In essence, relying on file-based features for ransomware detection is reactive. This implies that such methods either: (1) need the ransomware file to be stationary long enough for detection, or; (2) the ransomware has already been executed and encrypted files. Consequently, systems only employing this method are racing against time to safeguard other files from encryption. The effectiveness of this approach may diminish as computing power advances and encryption processes become faster and more efficient. The practical value of this strategy emerges when integrated into a real-time detection system that actively prevents further file encryption.

2) DATASETS USED THROUGHOUT LITERATURE FOR RANSOMWARE RESEARCH

One significant challenge researchers face in ransomware detection is the need for publicly available datasets for training and evaluating machine learning models. Despite calls from previous authors emphasising the importance of dataset sharing [14], [21], up to 96% of datasets used in the literature have yet to be made public [164]. This trend persists in recent literature on ransomware detection, where over 85% of studies constructed their datasets by collecting ransomware samples from repositories like VirusTotal[6] and VirusShare,[7] detonating them on secure servers, and extracting logs. Although these studies yield positive results, the lack of public dataset sharing hinders result validation and comparison. Reasons for not sharing datasets publicly range from labelling difficulties [147] to challenges related to dataset size, privacy concerns, and legal constraints [165]. As seen in Table.5, various datasets have been established in the literature to support ransomware research. These datasets include network traces of malware activity [138], [139], [163], as well as host-level captures such as system recordings [145], system calls [149], execution traces [151], and storage patterns [151]. While these datasets facilitate result comparisons, they are limited to the features they contain. Some authors have also created datasets comprising ransomware binaries [141], [147], [156], [159], allowing researchers to focus on feature extraction and ML model development with a shared baseline for result comparison.

[6]https://www.virustotal.com/
[7]https://virusshare.com/

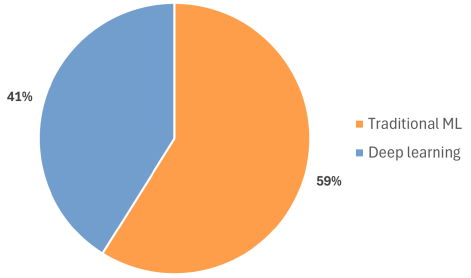

FIGURE 10. ML approaches used in surveyed studies.

However, researchers need to exercise caution to identify inactive ransomware samples over time.

3) MACHINE LEARNING ALGORITHMS

This section provides an overview of the machine learning (ML) algorithms employed in the literature for ransomware detection. The spectrum covers traditional ML algorithms like Support Vector Machines, Deep Learning algorithms such as Convolutional Neural Networks, and Ensemble techniques like Random Forest.

a: TRADITIONAL ML TECHNIQUES

Most studies reviewed have leveraged traditional machine learning (ML) to detect ransomware due to its ease of implementation and accurate results. As seen in Fig. 10, traditional ML techniques are the most commonly employed approach to detect ransomware throughout the literature. This includes algorithms such as Support Vector Machine (SVM), followed by decision trees and Naïve Bayes. Other algorithms less commonly used throughout the literature include linear regression. Most studies have experimented with various algorithms and compared the results to choose the most accurate classifier [102], [103]. The varying results can be attributed to the differences in features fed into the model and the datasets used.

In contrast to deep learning methods, traditional machine learning (ML) relies on domain experts to perform feature engineering and meticulously label data before feeding it into the machine learning model. Consequently, various studies in the literature distinguish themselves by employing diverse combinations of features and algorithms. Traditional ML techniques face the challenge of requiring constant updates to the underlying model due to the evolving nature of ransomware. This elevates the risk of a ransomware attack, especially given that traditional ML models are typically updated from scratch, and there may be a considerable time gap between updates. Another drawback of traditional machine learning is its difficulty handling intricate data structures and sequential patterns.

Overall, traditional ML offers ease of implementation, fast detection, low resource consumption, and accurate results at the expense of careful preparation and labelling of the data.

b: DEEP LEARNING TECHNIQUES

Traditional machine learning methods necessitate domain experts to label and extract meaningful features, making the process time-consuming. In contrast, deep learning approaches can handle raw data and extract significant features without relying on domain experts. Deep learning, a well-explored branch of machine learning, has demonstrated promising outcomes, especially with sequential data and visual object recognition. This is achieved by processing data across multiple layers and utilising general-purpose algorithms to derive higher levels of abstraction from previous layer outputs. Consequently, deep learning models can reveal unique insights that may be overlooked. The use of deep learning approaches within the ransomware detection domain is an emerging trend. Studies in the literature have employed deep learning algorithms like Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and Autoencoders to detect ransomware.

LSTM networks are frequently utilised in research due to their capacity to retain previously acquired knowledge. This characteristic proves particularly beneficial in ransomware detection, where the model needs to integrate new samples without discarding its existing knowledge. LSTMs excel in recognising patterns within sequential data, such as the time-series data commonly present in logs associated with ransomware detection. In their study, Karab et al. [115] applied LSTMs by representing run-time behavioural reports as sequences of words using word2vec, and inputting them into an LSTM network to identify malicious patterns.

CNNs are also widely employed, especially for their effectiveness in analysing visual data. Ciaramella et al. [124] employed CNNs to classify binary images as ransomware. They converted ransomware binaries into images, processed them through multiple convolution layers to extract features, such as combining neighbouring pixels, and then inputted them into a dense layer for classification. This approach enabled them to achieve an impressive 96.9% accuracy rate in ransomware detection.

Autoencoders have recently gained attention in the ransomware detection domain. As an artificial neural network for unsupervised learning, autoencoders encode input data into a compressed format and decode it back to the original data, minimising the difference between input and reconstructed output. Zahoora et al. [128] employed Contractive Autoencoders (CAEs) to extract host-based features from ransomware, encompassing API calls, registry key setups, and binary strings. The authors then used multiple classifiers to detect ransomware activity with high accuracy.

The utilisation of Generative Adversarial Networks (GANs) is an emerging area in ransomware detection research. GANs are designed to generate new data instances resembling a given dataset, proving highly valuable in adversarial learning tasks, such as creating scenarios for zero-day attacks where relevant data may be scarce. Several studies in the literature have employed GANs to generate synthetic

datasets [78], [117], augmenting real attack patterns due to the evolving nature of ransomware and the custom tactics involved in zero-day attacks.

However, despite the recent emphasis on deep learning techniques, it's noteworthy that these approaches tend to be resource-intensive and time-consuming. This can pose challenges, especially when aiming for real-time ransomware detection. Authors have proposed various solutions to address this limitation, including applying feature selection techniques to reduce the number of features [166] and improving classification speed through algorithm modifications [167].

While deep learning techniques present an exciting paradigm within the ransomware detection domain and show promise in detecting ransomware and generating synthetic attack scenarios, their resource demands should be carefully considered.

c: ENSEMBLE LEARNING TECHNIQUES

Ensemble learning models combine multiple base models to enhance detection accuracy. This approach compensates for errors in individual models by leveraging the strengths of others, resulting in improved overall accuracy. Ensemble models address common challenges in machine learning, including class imbalance, concept drift, and overfitting (often called the curse of dimensionality) [168].

Random forest is the most widely used algorithm in the ransomware detection domain due to its simplicity of implementation and high accuracy. Boosting is another frequently applied technique, with studies using Ada-Boost, Gradient Boosting, or XGBoost. Other studies have adopted a fusion approach, combining various algorithms and weighting their outputs [98].

Ensemble learning methods offer high accuracy and ease of implementation and have proven particularly valuable in early ransomware detection scenarios where limited data is available for classification [169].

d: HYBRID APPROACHES

Recent studies have embraced hybrid approaches involving different algorithms for distinct tasks. One such approach involves employing deep learning for feature extraction and subsequently utilising traditional machine learning algorithms for classification.

For instance, Zahoora et al. [128] combine Contractive Autoencoders (CAEs) with cost-sensitive base classifiers for ransomware detection with high accuracy. The authors use deep CAEs for unsupervised feature extraction, leveraging their ability to extract robust feature representations. The extracted features are then input into a cost-sensitive Pareto Ensemble of base classifiers, including cost-sensitive Support Vector Machine (SVM), weighted Logistic Regression (LR), and cost-sensitive Random Forest (RF). This hybrid approach allows the authors to derive features through unsupervised deep learning while benefiting from an ensemble of cost-sensitive traditional ML algorithms, maintaining good classification performance, and addressing class imbalance.

Gulmez et al. [113] incorporate Explainable Artificial Intelligence (XAI) with Convolutional Neural Networks (CNNs). While deep learning has proven to detect ransomware activity successfully, it operates as a black box, making it challenging for security professionals to comprehend the patterns leading to specific file classifications. XAI addresses this issue by highlighting the features or characteristics within the data that significantly influence these classifications. The authors initially used CNN to extract useful features and conduct classification in their study. Subsequently, they employ Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) XAI models to provide local and global explanations for the detection outcomes. This approach yields a high accuracy rate and offers a unique understanding of the features contributing most to detection.

Although hybrid approaches introduce an exciting new paradigm in the ransomware detection domain, further research is required to assess their real-world implications, particularly regarding the trade-offs between model complexity, resource consumption, and classification performance.

4) APPROACHES TO UPDATE THE ML MODEL

This section delineates the methodologies employed in the literature to update machine-learning models within the examined studies. The literature predominantly employs two methods: non-incremental approaches, which involve updating the machine-learning model from scratch, and incremental approaches, where the machine-learning model undergoes gradual updates over time.

a: NON-INCREMENTAL LEARNING APPROACHES

The predominant approach for training and testing machine-learning models in the literature is non-incremental. These studies typically involve executing ransomware samples on virtual machines, conducting feature engineering, and training and testing models in offline batches [72]. While this approach often produces accurate results with sufficient upfront training data, its drawback is the necessity to train the entire model from scratch when updates are required [170]. When encountering an unknown ransomware sample in the wild, retraining the model with all available data is imperative to guard against zero-day attacks. During the time window between model training, there is the potential for a zero-day attack. The escalating number of newly detected malware samples compounds this challenge. In 2023 alone, AV-Test recorded over 1.1 billion new malware samples, indicating a 10% increase from the previous year.[8] Consequently, retraining non-incremental learning models from scratch with the influx of new ransomware samples becomes an uphill battle that demands time and resources.

[8]https://www.av-test.org/en/statistics/malware

TABLE 6. The strengths and limitations of detection approaches and ML models used in ransomware detection systems.

		Strengths	Limitations
Detection approach	**Delayed detection**	- High accuracy due to the ability of training on the full dataset before classification occurs - Larger range of data to train ML models due to availability of more data - Low risk as it limits the exposure to ransomware from running on bare metal systems	- The file must be located first before it can be sent to the virtual machine for detonation. This may not be practical in real-world scenarios - Classification does not occur in real-time which may slow user operations down as they wait for feature extraction and classification
	Real-time detection	- Detection even if the ransomware file is not located - Last line of defence. Ransomware detection occurs as the ransomware is running - Ability to detect fileless malware (if using non file-based features) and more advanced attacks where the binary may not be present	- The underlying model is susceptible to concept drift due to the evolution of ransomware - Lower accuracy rate since there is less data available upfront for classification - Resource intensive if using Deep learning approaches - Higher margin of error and higher risk since there is less time for ransomware detection - Setup is more complicated since a synergy between feature extraction and ML models is needed
ML approach	**Traditional ML**	- Ease of implementation - High accuracy - Resource efficient - Control over feature engineering - Interpretability is easier than Deep learning approaches (easier to understand how the model arrived at a decision)	- Labelling of data required by domain experts - Struggles to capture long-term and sequential patterns - Susceptible to overfitting
	Deep learning	- High performance - Deep learning can learn features from raw data eliminating the need for complex feature engineering from domain experts - Excels in detecting sequential relationships and temporal dependencies in time-series data - Deep learning models are well-suited for detecting complex patterns such as anomalies	- Resource intensive - Slower training rate, which may create a vulnerability to zero-day ransomware - Difficulty in determining how the model arrived at a decision

b: INCREMENTAL LEARNING APPROACHES

Incremental learning, an alternative method that gradually updates the underlying machine learning model, has succeeded in domains like image classification. However, only a subset of studies have explored its utility in ransomware detection and classification.

Roy and Chen [83] addressed the challenge of evolving ransomware by incorporating incremental learning into their model, using bidirectional LSTM (Bi-LSTM) to update the model with new data. Since incremental learning models are continuously updated, there is potential for degradation to occur on the underlying model as the data evolves. This leads to a mismatch between training and real-world data, known as concept drift. To manage this, the authors employed backpropagation with gradient descent to calculate and minimise the loss function.

Al-Rimy et al. [169] introduced a novel incremental iBagging technique to update the dataset to mitigate the lack of sufficient information in the early phases of a ransomware attack. Features were then selected using the Enhanced Semi-Random Subspace (ESRS) technique, and classification occurred through an ensemble of algorithms. However, this approach is susceptible to drift due to the absence of a mechanism for minimising loss over time.

While these studies present positive results with incremental learning approaches, several challenges emerge when using this technique, including: (1) resource constraints during model training; (2) the risk of forgetting previously learned ransomware samples when introducing new models, and; (3) susceptibility to concept drift. Overall, incremental learning approaches offer a means to frequently update the underlying model without retraining it from scratch.

C. RANSOMWARE DETECTION SYSTEM DESIGN

This section explores the design considerations and architectures of ransomware detection systems proposed in the literature.

1) DETECTION APPROACHES

This section examines detection approaches presented in the literature, encompassing both delayed and real-time detection

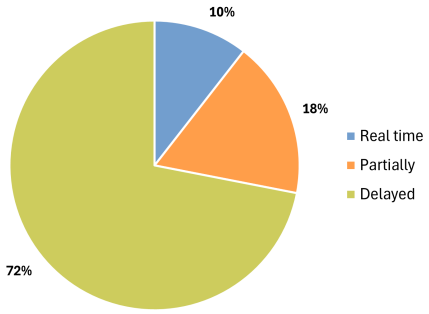

FIGURE 11. A segmented view of detection approaches used throughout the literature.

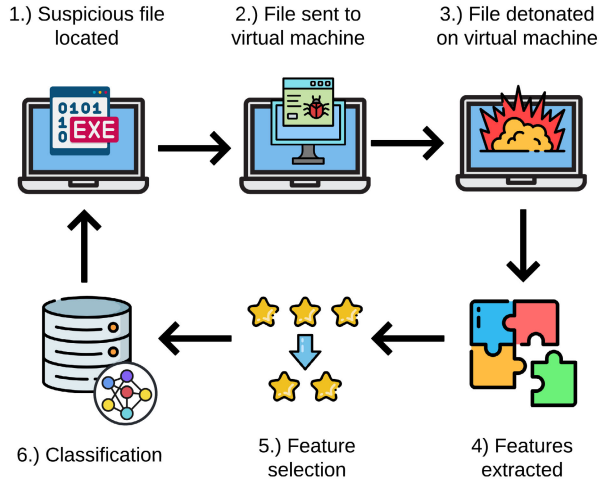

FIGURE 12. Ransomware detection using a delayed approach.

methods. Fig. 11 provides a segmented view of the detection approaches employed in the literature.

a: DELAYED DETECTION

Delayed detection, as shown in Fig. 12, involves detonating the ransomware sample on a secure virtual machine to extract the features required for classification. Delayed detection is the most common approach presented throughout the literature due to its ease of implementation and high detection accuracy since the classifying algorithm has more features. This approach increases the chance of finding ransomware while reducing the likelihood of encrypted sensitive data. However, the caveat of this approach is that suspicious files must first be intercepted to be detonated in a secured environment. Traditional ML algorithms can be leveraged, making detection less resource-intensive. Several studies have leveraged delayed detection approaches [12], [72].

Alqahtani and Sheldon [12] transfer suspicious executable files to a Cuckoo Sandbox for further analysis. The authors use Deep Belief Networks (DBN) to classify ransomware after extracting Cryptographic APIs and I/O request packets. Zuhair and Selamat [72] adopt a similar approach and extract dynamic features such as API calls, file operations and file settings after detonating the file in a secured sandbox.

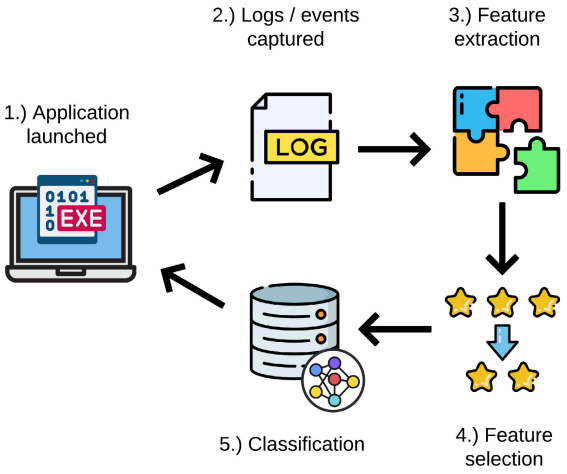

FIGURE 13. Ransomware detection approach in real-time.

Ransomware researchers have achieved impressive detection accuracies using delayed approaches. Still, there are several challenges, as can be seen in Table. 6, such as: (1) the suspicious file must be intercepted and detonated on a secured virtual machine. This is becoming more difficult as malware evasion techniques become more advanced, particularly since ransomware often has the awareness to detect if it is being analysed within a virtualised environment and prevents malicious payloads from being deployed, and; (2) the classification does not occur in real-time, which may slow the user experience.

b: REAL-TIME DETECTION

Some studies have successfully detected ransomware in real-time, the process of which can be seen in Fig 13. Real-time detection, distinct from dynamic analysis (extracting features at run-time for subsequent analysis), involves live feature extraction and classification from when a file is executed until encryption is initiated. Some studies use partial real-time approaches by intercepting the file in real-time and then using delayed detection methods for classification. For instance, Shaukat and Ribeiro [82] scan suspicious files for static "red flags" in real-time and transfer them to a virtualised environment for detonation. Keong et al. [98] adopt a similar method by intercepting executables passing through an IPS gateway. These suspicious files are then forwarded to a sandbox environment for detonation and subsequent classification.

Other studies propose complete real-time ransomware detection systems whereby the collection of data, feature extraction and classification occur in near real-time. Roy and Chen [83] adopt this approach in their study and model run-time event sequences from the Windows Logging Service (WLS) with BiLSTM-CLF. This approach detects anonymous events caused by ransomware and achieves an impressive detection rate of 99.87%. Unlike delayed methods, the authors argue that this approach enables

FIGURE 14. Ransomware detection using a centralised architecture.

FIGURE 15. Ransomware detection using a federated architecture.

them to detect ransomware on bare-metal servers. Deep learning has demonstrated strong real-time accuracy in detecting malicious sequences, particularly when leveraging algorithms such as LSTM. For instance, Homayoun et al. [80] capture sequences of syscalls 10 seconds after launching applications and classify the events with LSTM, achieving a 97.20% detection accuracy.

While real-time detection has shown positive results, several challenges are evident, including: (1) susceptibility to concept drift; (2) the inherent limitation of having less data available for classification compared to delayed detection approaches. This makes early detection in real-time challenging and riskier (especially if detection occurs on bare-metal systems), and; (3) the higher maintenance cost of deep learning approaches due to their resource-intensive training, often requiring systems with multiple CPUs or GPUs in a centralised setup [80]. Overall, several studies have demonstrated the capability of detecting ransomware in real-time.

2) SYSTEM ARCHITECTURE

This section explores the architectural approaches employed by studies in the literature, encompassing both centralised and federated architectures.

a: CENTRALISED ARCHITECTURE

Most machine learning-based ransomware detection systems are designed with a centralised architecture, as shown in Fig. 14. This means the model-building and classification functions are centralised. This architecture is typically presented in two configurations in the literature: (1) feature extraction and classification occur on the same system, and (2) features are extracted on client systems and then sent to an external server for classification. Both configurations have been extensively discussed [24]. Most studies handle feature engineering, and classification occurs on the same system. To address security concerns related to detonating ransomware on bare-metal servers, researchers often execute ransomware within virtual machines and extract the logs for analysis [72]. This approach is considered safer, as ransomware can be isolated within the virtual machine and removed afterwards. Following detonation, features are often forwarded to the ML model for classification internally within the virtual machine or to the host [24].

Roy and Chen adopt an alternative approach [83], and send raw data to a centralised Linux Syslog server for further processing and classification. In their study, the authors run ransomware on bare-metal servers, capturing event ID sequences from Windows event logs, which are then forwarded to an external server on the network for classification. This architecture enables the pooling of hardware resources, allowing the use of more complex algorithms, such as deep learning, that otherwise may not be practical if installed individually on endpoints due to resource constraints.

The challenges associated with centralised architectures include: (1) in a highly interconnected network, the process of extracting features, sending logs to an external host, and performing classification introduces a time delay; (2) availability concerns, as an attack on the classifying server could hinder ransomware detection, and (3) there is potential for sensitive data to be intercepted otherwise known as man-in-the-middle (MITM) attacks. Overall, centralised architectures are the most commonly used setup for ransomware detection throughout the literature.

b: FEDERATED ARCHITECTURE

An alternative approach to ransomware detection involves a federated architecture as shown in Fig. 15. In contrast to centralised architectures, where the machine learning model is constructed on a single server, a federated architecture distributes the model's training across multiple nodes, aggregating the results to enhance detection accuracy. This approach offers several advantages: (1) distributed architectures eliminate single points of failure, enhancing system availability during the training and testing of the machine learning model; (2) nodes within the network can share their insights into ransomware, fostering the potential for improved detection accuracy; (3) with privacy regulations like the General Data Protection Regulation (GDPR), protecting personal information (PI) is crucial (refer to Table 1). Adopting a federated approach involves sending

only the model, not the raw data, during model training, enhancing security practices.

While few studies explore this type of setup for ransomware detection, Thapa et al. [171] utilised federated learning for training and testing their machine learning models. The authors designed a privacy-preserving ransomware detection system for hospitals, considering the distributed nature of healthcare data storage and the need for data sharing among hospitals.

In this federated architecture, a supernode sends an aggregated machine learning model (global model) at time t to each node (e.g., hospital) in the network. Individual nodes then train and test their models using local data. After this stage, each model is returned to the supernode for aggregation into a global model. This ensures that only the machine learning model is transmitted, preserving the privacy of PI data and mitigating man-in-the-middle attacks (MITM). While this approach aligns with the authors' use case, it presents challenges, including (1) the risk of poisoning the centralised model due to aggregation from individual nodes and; (2) the potential for a model update gap. The time delay between updating the global and local models creates a window of opportunity for ransomware to impact nodes that have yet to be updated by the global model.

V. LIMITATIONS AND FUTURE DIRECTIONS

In this section, we elucidate the constraints identified in current studies and present anticipated future directions derived from discernible trends in the literature. The objective is to guide future research endeavours within ransomware detection.

A. LIMITATIONS OF EXISTING STUDIES

1) INCONSISTENCY AND LACK OF COHERENCE IN REPORTING RESULTS

Numerous studies throughout the literature have successfully identified ransomware activity with impressive detection accuracies. However, there is a notable lack of consistency in reporting these results. Some studies exclusively present detection accuracies [167], [172], while others focus on precision and recall, disregarding additional metrics [87], [89]. Meanwhile, others only report false positives [173]. Moreover, a limited number of studies provide insight into the rationale behind their metric selection. Simply reporting metrics such as accuracy may not provide a full picture, particularly in anomaly detection, where malicious activity constitutes a minority class. In these cases, accuracy alone may not accurately reflect the efficacy of ransomware detection. This consideration extends to metrics like false positives, where a false negative could be more detrimental than a false positive in ransomware detection scenarios. The inconsistency in reported results across studies complicates comparisons and diminishes transparency in evaluating ransomware detection methods.

2) VALIDATION OF PREVIOUS STUDIES

Several studies gauge the effectiveness of their methodologies by comparing metrics reported in other studies. However, this method poses challenges to validating previous findings due to variations in machine learning algorithms, features, and datasets among studies. Moreover, the ongoing evolution of malware and advancements in operating system security features cast doubt on the relevance of previous findings and their applicability in the current landscape. For instance, some studies use legacy datasets or employ older versions of Windows for detonating ransomware samples, which may not reflect the effectiveness of malware on newer Windows versions with updated security measures. Validating the results of prior research can provide insights into malware evolution and the relevance of earlier methodologies.

3) LACK OF FOCUS ON REAL-TIME DETECTION

While a substantial body of research on ransomware detection exists, most studies concentrate on enhancing the detection accuracy of machine learning models without considering their practical application in real-world scenarios. As a result, numerous studies utilise "delayed detection" techniques to identify malicious behaviour. This method entails detonating ransomware samples within a secure virtual environment for a specific duration, extracting features, and subsequently inputting the data into an ML model for classification. This approach is frequently adopted due to its straightforward implementation and high accuracy, often employing traditional machine learning models such as SVM. However, achieving real-time detection with this approach is challenging because ransomware samples must be fully detonated before extracting relevant features. In environments with sensitive data, this poses a risk as files may become encrypted before classification begins.

4) RESOURCE CONSTRAINTS

Previous studies primarily relied on traditional machine learning algorithms for training and testing due to their ease of implementation and reliable performance. However, there is a growing interest in utilising deep learning for ransomware detection, offering superior accuracy and potential for real-time detection. Nonetheless, deep learning models are computationally demanding and costly to deploy [174]. Some studies have addressed this challenge by deploying their models in environments with ample CPU cores or GPUs [175]. Consequently, most deep learning systems proposed so far are confined to centralised servers capable of handling the substantial hardware resources needed for training and testing.

5) MACHINE LEARNING MODELS NOT UPDATED

Most studies in the literature operate in batch mode without incorporating a mechanism for updating their machine-learning models. Given the constant evolution of malware,

it's essential to regularly update these models with the latest variants to ensure effective threat detection. However, the prevalent approach in ransomware detection proposals involves updating the underlying model from scratch. This time-consuming process leaves a window of opportunity for new malware variants to evade detection during the update period.

B. FUTURE DIRECTIONS

1) REPORTING WITH INTENTION

As mentioned, many studies fail to select their reporting metrics carefully and instead rely on standard metrics like Accuracy and F1 score without discussing their advantages and limitations. This focus on competing with other studies on metrics like accuracy may not provide a complete understanding of the experiment's constraints. We encourage future researchers to include a variety of metrics that shed light on the study's limitations, such as false positives, false negatives, and the Matthews correlation coefficient (MCC). Researchers need to deliberate on their chosen metrics. For instance, metrics like the area under the receiver operating characteristic (AUC ROC) can exhibit high variability when applied to imbalanced datasets [176]. The Matthews Correlation Coefficient (MCC) stands out for its robustness against imbalanced datasets and is favoured as the metric of choice. Unlike other metrics, such as AUC ROC, a high MCC consistently corresponds to a high AUC ROC, but the reverse is not always true [177]. Therefore, researchers should carefully consider their needs and justify their metric choices to enhance transparency. Future researchers are also encouraged to provide comprehensive reports on the performance metrics of their experiments, including detection timings. This will help understand the potential trade-offs associated with using a particular technique.

2) SYNTHETIC DATASETS

As ransomware continues to evolve, many studies rely on outdated samples or datasets, making ransomware research primarily reactionary. Keeping datasets up to date presents challenges because live samples shared on public repositories, like VirusShare, have a limited lifespan due to disabled C&C servers. Synthetic datasets offer a solution to this challenge by enabling ransomware researchers to concentrate on improving ransomware detection systems rather than the laborious task of assembling a collection of active ransomware to build a dataset. Researchers should aim to develop synthetic datasets that are regularly updated, providing valuable resources for security researchers. Additionally, using synthetic datasets standardises detection results across studies, facilitating baseline comparisons.

3) REAL-TIME DETECTION APPROACHES

As observed in the survey, there is a noticeable lack of focus on creating ransomware detection systems suitable for real-world scenarios, particularly in real-time detection. Researchers should prioritise designing ransomware detection systems that are applicable in real-world settings. This involves exploring real-time detection methods that don't require virtual machines to detonate ransomware before classification. Although deep learning approaches show potential in this area, more work is needed to streamline feature extraction techniques and improve detection speed. Researchers could utilise the insights from this survey to integrate features from different phases of the attack lifecycle (such as those for identifying ransomware delivery and communication with C&C servers) to lessen the dependence on real-time detection at the host level. Finally, future researchers should also include reporting on the time taken to detect ransomware activity, demonstrating the practical utility of their models in real-time ransomware detection scenarios.

4) EXPLORE AGENT-BASED, INCREMENTAL AND TRANSFER LEARNING APPROACHES

While transfer learning and incremental learning have succeeded in domains like healthcare and traffic control [178], their potential in ransomware detection remains largely unexplored. Transfer learning could aid in detecting previously unseen malware families, reducing the reliance on extensive training datasets. Similarly, incremental learning offers promise by allowing algorithms to learn gradually, reducing the need for full model retraining and saving time and resources. Additionally, the application of multi-agent systems in ransomware detection has yet to be thoroughly investigated. While multi-agent systems have been studied in intrusion detection [179], exploring the use of intelligent agents collaborating within the ransomware domain presents an intriguing avenue for further research.

5) ADVERSARIAL LEARNING

While some studies have addressed adversarial learning and its potential threats [34], [180], this area remains relatively unexplored within the ransomware detection domain. As ransomware detection methods continue to advance, it's expected that future ransomware will exploit adversarial learning techniques to evade detection by mimicking benign behaviours learned from machine learning models. Therefore, it is recommended that research efforts in adversarial learning be intensified, focusing on its application in generating samples with tailored behaviours and developing techniques to effectively detect ransomware produced through adversarial learning.

VI. CONCLUSION

In this survey, we thoroughly reviewed ransomware detection studies in the literature. Our pragmatic approach involved a comprehensive examination of ransomware detection system designs. Throughout this review, we covered various aspects, including recent ransomware families observed in the wild, government responses, available datasets, classification based on detection timing, and machine learning techniques used, and discussed limitations and future recommendations.

Despite ransomware's rapid evolution, many proposed detection systems lack practicality for real-world ransomware attacks. As a result, most systems are reactive and involve delayed detection. To effectively mitigate the ransomware threat, researchers must design detection systems capable of swiftly identifying ransomware early on while addressing the evasion techniques employed by modern malware.

REFERENCES

[1] J. Ispahany and R. Islam, "Detecting malicious COVID-19 URLs using machine learning techniques," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Other Affiliated Events (PerCom Workshops)*, Mar. 2021, pp. 718–723.

[2] (Nov. 2021). *Treasury Continues to Counter Ransomware As Part of Whole-of-Government Effort; Sanctions Ransomware Operators and Virtual Currency Exchange*. [Online]. Available: https://home.treasury.gov/news/press-releases/jy0471

[3] Chainalysis. (2023). *The 2023 Crypto Crime Report*. [Online]. Available: https://go.chainalysis.com/rs/503-FAP-074/images/Crypto_Crime_Report_2023.pdf

[4] (Aug. 2020). *Australian Broadband Data Demand: Data Demand on the Nbn Continues to Reflect High Network Usage*. [Online]. Available: https://www.nbnco.com.au/corporate-information/media-centre/media-statements/data-demand-continues-to-reflect

[5] (2020). *Gartner CFO Survey: 74% to Shift Some Employees to Remote Work Permanently*. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2020-04-03-gartner-cfo-survey-reveals-74-percent-of-orgs-to-shift-some-employees-to-remote-work-permanently

[6] F. Yarochkin. (Oct. 2021). *Ransomware As a Service: Enabler of Widespread Attacks*. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ransomware-as-a-service-enabler-of-widespread-attacks

[7] J. Beerman, D. Berent, Z. Falter, and S. Bhunia, "A review of colonial pipeline ransomware attack," in *Proc. IEEE/ACM 23rd Int. Symp. Cluster, Cloud Internet Comput. Workshops (CCGridW)*, May 2023, pp. 8–15.

[8] J. Dossett. (Nov. 2021). *A Timeline of the Biggest Ransomware Attacks*. [Online]. Available: https://www.cnet.com/personal-finance/crypto/a-timeline-of-the-biggest-ransomware-attacks/

[9] M. Burgess. (Jun. 2022). *Conti's Attack Against Costa Rica Sparks a New Ransomware Era*. [Online]. Available: https://www.wired.com/story/costa-rica-ransomware-conti/

[10] R. Falk and A.-L. Brown. (2021). *Underwritten or Oversold?—Cyber Security CRC*. [Online]. Available: https://cybersecuritycrc.org.au/sites/default/files/2021-10/Underwritten%20or%20oversold%20%20-%20DV.pdf

[11] F. Aldauiji, O. Batarfi, and M. Bayousef, "Utilizing cyber threat hunting techniques to find ransomware attacks: A survey of the state of the art," *IEEE Access*, vol. 10, pp. 61695–61706, 2022.

[12] A. Alqahtani and F. T. Sheldon, "A survey of crypto ransomware attack detection methodologies: An evolving outlook," *Sensors*, vol. 22, no. 5, p. 1837, Feb. 2022.

[13] A. Alraizza and A. Algarni, "Ransomware detection using machine learning: A survey," *Big Data Cognit. Comput.*, vol. 7, no. 3, p. 143, Aug. 2023.

[14] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Comput. Secur.*, vol. 74, pp. 144–166, May 2018.

[15] 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.

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

[17] N. M. Chayal, A. Saxena, and R. Khan, "A review on spreading and forensics analysis of windows-based ransomware," *Ann. Data Sci.*, pp. 1–22, Jun. 2022.

[18] 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.

[19] J. A. G. Hernández, P. G. Teodoro, R. M. Carrión, and R. R. Gómez, "Crypto-ransomware: A revision of the state of the art, advances and challenges," *Electronics*, vol. 12, no. 21, p. 4494, Nov. 2023.

[20] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, "Ransomware detection, avoidance, and mitigation scheme: A review and future directions," *Sustainability*, vol. 14, no. 1, p. 8, Dec. 2021.

[21] A. M. Maigida, S. M. Abdulhamid, M. Olalere, J. K. Alhassan, H. Chiroma, and E. G. Dada, "Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms," *J. Reliable Intell. Environments*, vol. 5, no. 2, pp. 67–89, Jul. 2019.

[22] T. McIntosh, A. S. M. Kayes, Y.-P.-P. Chen, A. Ng, and P. Watters, "Ransomware mitigation in the modern era: A comprehensive review, research challenges, and future directions," *ACM Comput. Surveys*, vol. 54, no. 9, pp. 1–36, Dec. 2022.

[23] R. Moussaileb, N. Cuppens, J.-L. Lanet, and H. L. Bouder, "A survey on windows-based ransomware taxonomy and detection mechanisms," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–36, Jul. 2022.

[24] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—A state of the art survey," *ACM Comput. Surveys*, vol. 52, no. 5, pp. 1–48, Sep. 2020.

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

[26] N. Rani, S. V. Dhavale, A. Singh, and A. Mehra, "A survey on machine learning-based ransomware detection," in *Proc. 7th Int. Conf. Math. Comput. (ICMC)*. Springer, 2021, pp. 171–186. [Online]. Available: https://link.springer.com/book/10.1007/978-981-16-6890-6?page=1#toc

[27] S. Razaulla, C. Fachkha, C. Markarian, A. Gawanmeh, W. Mansoor, B. C. M. Fung, and C. Assi, "The age of ransomware: A survey on the evolution, taxonomy, and research directions," *IEEE Access*, vol. 11, pp. 40698–40723, 2023.

[28] D. Smith, S. Khorsandroo, and K. Roy, "Machine learning algorithms and frameworks in ransomware detection," *IEEE Access*, vol. 10, pp. 117597–117610, 2022.

[29] J. Singh and J. Singh, "A survey on machine learning-based malware detection in executable files," *J. Syst. Archit.*, vol. 112, Jan. 2021, Art. no. 101861.

[30] V. Thangapandian, "Machine learning in automated detection of ransomware: Scope, benefits and challenges," in *Illumination of Artificial Intelligence in Cybersecurity and Forensics*. Springer, 2022, pp. 345–372. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-93453-8

[31] U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," *Appl. Sci.*, vol. 12, no. 1, p. 172, Dec. 2021.

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

[33] N. Hampton, Z. Baig, and S. Zeadally, "Ransomware behavioural analysis on windows platforms," *J. Inf. Secur. Appl.*, vol. 40, pp. 44–51, Jun. 2018.

[34] L. Caviglione, M. Choras, I. Corona, A. Janicki, W. Mazurczyk, M. Pawlicki, and K. Wasielewska, "Tight arms race: Overview of current malware threats and trends in their detection," *IEEE Access*, vol. 9, pp. 5371–5396, 2021.

[35] The Federal Bureau of Investigation (FBI), ISA (CISA), The Multi-State Information Sharing, and AC (MS-ISAC). (2023). *Stopransomware: Rhysida Ransomware*. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-11/aa23-319a-stopransomware-rhysida-ransomware_1.pdf

[36] TFB of Investigation (FBI), The Cybersecurity, and ISA (CISA). (2022). *Stopransomware: Alphv Blackcat*. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-12/aa23-353a-stopransomware-alphv-blackcat_0.pdf

[37] ASCS. (2023). *Understanding Ransomware Threat Actors: Lockbit*. [Online]. Available: https://www.cyber.gov.au/about-us/advisories/understanding-ransomware-threat-actors-lockbit

[38] S. Gatlan. (Oct. 2021). *Accenture Confirms Data Breach After August Ransomware Attack*. [Online]. Available: https://www.bleepingcomputer.com/news/security/accenture-confirms-data-breach-after-august-ransomware-attack/

[39] L. Abrams. (Jun. 2022). *Lockbit 3.0 Introduces the First Ransomware Bug Bounty Program*. [Online]. Available: https://www.bleepingcomputer.com/news/security/lockbit-30-introduces-the-first-ransomware-bug-bounty-program/
[40] FB Investigation (FBI) and USSS. (2022). *Indicators of Compromise Associated With Blackbyte Ransomware*. [Online]. Available: https://www.ic3.gov/Media/News/2022/220211.pdf
[41] FB Investigation (FBI), Cybersecurity, and ISA. (2022). *Indicators of Compromise Associated With Avoslocker Ransomware*. [Online]. Available: https://www.ic3.gov/Media/News/2022/220318.pdf
[42] The Federal Bureau of Investigation (FBI), ISA (CISA), D Health, and HHS. (2022). *Stopransomware: Hive Ransomware*. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-321a
[43] S. Gatlan. (Jun. 2022). *Costa Rica's Public Health Agency Hit By Hive Ransomware*. [Online]. Available: https://www.bleepingcomputer.com/news/security/costa-rica-s-public-health-agency-hit-by-hive-ransomware/
[44] I. Ilascu. (Aug. 2021). *Hive Ransomware Attacks Memorial Health System, Steals Patient Data*. [Online]. Available: https://www.bleepingcomputer.com/news/security/hive-ransomware-attacks-memorial-health-system-steals-patient-data/
[45] Cybersecurity, ISA (CISA), and FB Investigation (FBI). (Jul. 2021). *Darkside Ransomware: Best Practices for Preventing Bus. Disruption From Ransomware Attacks*. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-131a
[46] S. B. Shimol. (Apr. 2022). *Return of the Darkside: Analysis of a Large-scale Data Theft Campaign*. [Online]. Available: https://www.varonis.com/blog/darkside-ransomware
[47] M. Schwirtz and N. Perlroth. (May 2021). *Darkside, Blamed for Gas Pipeline Attack, Says It is Shutting Down*. [Online]. Available: https://www.nytimes.com/2021/05/14/business/darkside-pipeline-hack.html
[48] Y. K. Bin Mohamed Yunus and S. Bin Ngah, "Ransomware: Stages, detection and evasion," in *Proc. Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manage. (ICSECS-ICOCSIM)*, Aug. 2021, pp. 227–231.
[49] S. Gatlan. (Oct. 2021). *Ukraine Arrests Clop Ransomware Gang Members, Seizes Servers*. [Online]. Available: https://www.bleepingcomputer.com/news/security/ukraine-arrests-clop-ransomware-gang-members-seizes-servers
[50] NI Standards and T (NIST). (Jun. 2023). *Cve-2023-0669 Detail*. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-0669
[51] Cybersecurity, ISA (CISA), and FB Investigation (FBI). (2023). *Stopransomware: Cl0p Ransomware Gang Exploits Cve-2023-34362 Moveit Vulnerability*. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-158a
[52] A. Mundo. (Jan. 2020). *Clop Ransomware*. [Online]. Available: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/clop-ransomware/
[53] ASCS. (2022). *Annual Cyber Threat Report, July 2021 to June 2022*. [Online]. Available: https://www.cyber.gov.au/sites/default/files/2023-06/Understanding-Ransomware-Threat-Actors_LockBit.pdf
[54] B. Toulas. (Mar. 2022). *Revil Ransomware Member Extradited to U.S. to Stand Trial for Kaseya Attack*. [Online]. Available: https://www.bleepingcomputer.com/news/security/revil-ransomware-member-extradited-to-us-to-stand-trial-for-kaseya-attack/
[55] ASCS. (Jul. 2021). *Kaseya Vsa Supply-chain Ransomware Attack*. [Online]. Available: https://www.cyber.gov.au/about-us/alerts/kaseya-vsa-supply-chain-ransomware-attack
[56] E. Millington. (Aug. 2020). *Revil*. [Online]. Available: https://attack.mitre.org/software/S0496/
[57] (Dec. 2021). *Ransomware Spotlight: Conti*. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/ransomware-spotlight/ransomware-spotlight-conti
[58] Cybersecurity and ISA. (Mar. 2022). *Conti Ransomware*. [Online]. Available: https://www.cisa.gov/news-events/alerts/2021/09/22/conti-ransomware
[59] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (STIX)," *Mitre Corp.*, vol. 11, pp. 1–22, Jan. 2012.
[60] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto, "A cyber-kill-chain based taxonomy of crypto-ransomware features," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 4, pp. 277–305, Dec. 2019.
[61] L. Abrams. (May 2020). *Lockbit Ransomware Self-Spreads to Quickly Encrypt 225 Systems*. [Online]. Available: https://www.bleepingcomputer.com/news/security/lockbit-ransomware-self-spreads-to-quickly-encrypt-225-systems/
[62] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 646–656, Mar. 2013.
[63] D. C. D'Elia, L. Invidia, and L. Querzoni, "Rope: Covert multi-process malware execution with return-oriented programming," in *Proc. Eur. Symp. Res. Comput. Secur.* Springer, 2021, pp. 197–217. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-88418-5
[64] S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A survey on malware analysis and mitigation techniques," *Comput. Sci. Rev.*, vol. 32, pp. 1–23, May 2019.
[65] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, "Malware dynamic analysis evasion techniques: A survey," *ACM Comput. Surveys*, vol. 52, no. 6, pp. 1–28, Nov. 2020.
[66] Sudhakar and S. Kumar, "An emerging threat fileless malware: A survey and research challenges," *Cybersecurity*, vol. 3, no. 1, pp. 1–12, Dec. 2020.
[67] (Mar. 2022). *Indicators of Compromise Associated With Avoslocker Ransomware*. [Online]. Available: https://www.ic3.gov/Media/News/2022/220318.pdf
[68] F. Tang, B. Ma, J. Li, F. Zhang, J. Su, and J. Ma, "RansomSpector: An introspection-based approach to detect crypto ransomware," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101997.
[69] P. Bajpai and R. Enbody, "Attacking key management in ransomware," *IT Prof.*, vol. 22, no. 2, pp. 21–27, Mar. 2020.
[70] M. Keshavarzi and H. R. Ghaffary, "I2CE3: A dedicated and separated attack chain for ransomware offenses as the most infamous cyber extortion," *Comput. Sci. Rev.*, vol. 36, May 2020, Art. no. 100233.
[71] (Feb. 2022). *Ransomware Spotlight: Lockbit*. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/ransomware-spotlight/ransomware-spotlight-lockbit
[72] H. Zuhair and A. Selamat, "Rands: A machine learning-based anti-ransomware tool for windows platforms," in *Advancing Technology Industrialization Through Intelligent Software Methodologies, Tools and Techniques*. Amsterdam, The Netherlands: IOS Press, 2019, pp. 573–587.
[73] H. Liu and P. Patras, "NetSentry: A deep learning approach to detecting incipient large-scale network attacks," *Comput. Commun.*, vol. 191, pp. 119–132, Jul. 2022.
[74] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic," *Expert Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118299.
[75] L. F. Maimó, A. H. Celdrán, Á. P. Gómez, F. G. 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.
[76] J. Singh, K. Sharma, M. Wazid, and A. K. Das, "SINN-RD: Spline interpolation-envisioned neural network-based ransomware detection scheme," *Comput. Electr. Eng.*, vol. 106, Mar. 2023, Art. no. 108601.
[77] A. O. Almashhadani, D. Carlin, M. Kaiiali, and S. Sezer, "MFMCNS: A multi-feature and multi-classifier network-based system for ransomworm detection," *Comput. Secur.*, vol. 121, Oct. 2022, Art. no. 102860.
[78] 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.
[79] A. Buriro, A. B. Buriro, T. Ahmad, S. Buriro, and S. Ullah, "MalwD&C: A quick and accurate machine learning-based approach for malware detection and categorization," *Appl. Sci.*, vol. 13, no. 4, p. 2508, Feb. 2023.
[80] 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.
[81] M. E. Ahmed, H. Kim, S. Camtepe, and S. Nepal, "Peeler: Profiling kernel-level events to detect ransomware," in *Proc. Eur. Symp. Res. Comput. Secur.* Springer, 2021, pp. 240–260. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-88418-5
[82] 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)*, 2018, pp. 356–363.
[83] 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.

[84] A. O. Almashhadani, M. Kaiiali, D. Carlin, and S. Sezer, "MaldomDetector: A system for detecting algorithmically generated domain names with machine learning," *Comput. Secur.*, vol. 93, Jun. 2020, Art. no. 101787.
[85] Q. A. Al-Haija and A. A. Alsulami, "High performance classification model to identify ransomware payments for heterogeneous Bitcoin networks," *Electronics*, vol. 10, no. 17, p. 2113, Aug. 2021.
[86] S. A. Alsaif, "Machine learning-based ransomware classification of Bitcoin transactions," *Appl. Comput. Intell. Soft Comput.*, vol. 2023, pp. 1–10, Jan. 2023.
[87] 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.
[88] S. Poudyal and D. Dasgupta, "Analysis of crypto-ransomware using ML-based multi-level profiling," *IEEE Access*, vol. 9, pp. 122532–122547, 2021.
[89] S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, "On the classification of microsoft-windows ransomware using hardware profile," *PeerJ Comput. Sci.*, vol. 7, p. e361, Feb. 2021.
[90] 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 Manage.*, vol. 19, no. 1, pp. 19–36, Mar. 2022.
[91] Y. A. Ahmed, B. Koçer, S. Huda, B. A. Saleh 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.
[92] S. Sharmeen, Y. A. Ahmed, S. Huda, B. S. Koçer, 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.
[93] 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.
[94] 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.
[95] B. M. Khammas, "Ransomware detection using random forest technique," *ICT Exp.*, vol. 6, no. 4, pp. 325–331, Dec. 2020.
[96] 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.
[97] S. I. Bae, G. B. Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," *Concurrency Computation: Pract. Exper.*, vol. 32, no. 18, p. e5422, Sep. 2020.
[98] C. Keong Ng, S. Rajasegarar, L. Pan, F. Jiang, and L. Y. Zhang, "VoterChoice: A ransomware detection honeypot with multiple voting framework," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 14, Jul. 2020, Art. no. e5726.
[99] 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.
[100] 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.
[101] 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.
[102] A. Cohen and N. Nissim, "Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory," *Expert Syst. Appl.*, vol. 102, pp. 158–178, Jul. 2018.
[103] N. Nissim, Y. Lapidot, A. Cohen, and Y. Elovici, "Trusted system-calls analysis methodology aimed at detection of compromised virtual machines using sequential mining," *Knowl.-Based Syst.*, vol. 153, pp. 147–175, Aug. 2018.
[104] M. S. Abbasi, H. Al-Sahaf, M. Mansoori, and I. Welch, "Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection," *Appl. Soft Comput.*, vol. 121, May 2022, Art. no. 108744.
[105] M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan, "Ransomware classification and detection with machine learning algorithms," in *Proc. IEEE 12th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2022, pp. 0316–0322.
[106] R. Chaganti, V. Ravi, and T. D. Pham, "A multi-view feature fusion approach for effective malware classification using deep learning," *J. Inf. Secur. Appl.*, vol. 72, Feb. 2023, Art. no. 103402.
[107] S. Li, Y. Li, X. Wu, S. A. Otaibi, and Z. Tian, "Imbalanced malware family classification using multimodal fusion and weight self-learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7642–7652, Jul. 2022.
[108] I. Ba'abbad and O. Batarfi, "Proactive ransomware detection using extremely fast decision tree (EFDT) algorithm: A case study," *Computers*, vol. 12, no. 6, p. 121, Jun. 2023.
[109] C. Woralert, C. Liu, and Z. Blasingame, "HARD-lite: A lightweight hardware anomaly realtime detection framework targeting ransomware," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 12, pp. 5036–5047, Dec. 2023.
[110] X. Deng, M. Cen, M. Jiang, and M. Lu, "Ransomware early detection using deep reinforcement learning on portable executable header," *Cluster Comput.*, vol. 27, no. 2, pp. 1867–1881, Apr. 2024.
[111] 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.
[112] K. Thummapudi, P. Lama, and R. V. Boppana, "Detection of ransomware attacks using processor and disk usage data," *IEEE Access*, vol. 11, pp. 51395–51407, 2023.
[113] S. Gulmez, A. Gorgulu Kakisim, and I. Sogukpinar, "XRan: Explainable deep learning-based ransomware detection using dynamic analysis," *Comput. Secur.*, vol. 139, Apr. 2024, Art. no. 103703.
[114] D. W. Fernando and N. Komninos, "FeSAD ransomware detection framework with machine learning using adaption to concept drift," *Comput. Secur.*, vol. 137, Feb. 2024, Art. no. 103629.
[115] E. B. Karbab, M. Debbabi, and A. Derhab, "SwiftR: Cross-platform ransomware fingerprinting using hierarchical neural networks on hybrid features," *Expert Syst. Appl.*, vol. 225, Sep. 2023, Art. no. 120017.
[116] P. M. Anand, P. V. S. Charan, and S. K. Shukla, "HiPeR–early detection of a ransomware attack using hardware performance counters," *Digit. Threats, Res. Pract.*, vol. 4, no. 3, pp. 1–24, Sep. 2023.
[117] M. Gazzan and F. T. Sheldon, "An enhanced minimax loss function technique in generative adversarial network for ransomware behavior prediction," *Future Internet*, vol. 15, no. 10, p. 318, Sep. 2023.
[118] 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.
[119] A. Alqahtani and F. T. Sheldon, "Temporal data correlation providing enhanced dynamic crypto-ransomware pre-encryption boundary delineation," *Sensors*, vol. 23, no. 9, p. 4355, Apr. 2023.
[120] A. Singh, Z. Mushtaq, H. A. Abosaq, S. N. F. Mursal, M. Irfan, and G. Nowakowski, "Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data," *Electronics*, vol. 12, no. 18, p. 3899, Sep. 2023.
[121] M. A. Ayub, A. Siraj, B. Filar, and M. Gupta, "RWArmor: A static-informed dynamic analysis approach for early detection of cryptographic windows ransomware," *Int. J. Inf. Secur.*, vol. 23, no. 1, pp. 533–556, Feb. 2024.
[122] C. B N and B. S H, "Revolutionizing ransomware detection and criticality assessment: Multiclass hybrid machine learning and semantic similarity-based end2end solution," *Multimedia Tools Appl.*, vol. 83, no. 13, pp. 39135–39168, Oct. 2023.
[123] G. O. Ganfure, C.-F. Wu, Y.-H. Chang, and W.-K. Shih, "RTrap: Trapping and containing ransomware with machine learning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1433–1448, 2023.
[124] G. Ciaramella, G. Iadarola, F. Martinelli, F. Mercaldo, and A. Santone, "Explainable ransomware detection with deep learning techniques," *J. Comput. Virol. Hacking Techn.*, vol. 20, no. 2, pp. 317–330, Sep. 2023.
[125] G. O. Ganfure, C.-F. Wu, Y.-H. Chang, and W.-K. Shih, "DeepWare: Imaging performance counters with deep learning to detect ransomware," *IEEE Trans. Comput.*, vol. 72, no. 3, pp. 600–613, Mar. 2023.
[126] M. Wazid, A. Kumar Das, and S. Shetty, "BSFR-SH: Blockchain-enabled security framework against ransomware attacks for smart healthcare," *IEEE Trans. Consum. Electron.*, vol. 69, no. 1, pp. 18–28, Feb. 2023.
[127] Prachi and 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.

[128] U. Zahoora, A. Khan, M. Rajarajan, S. H. Khan, M. Asam, and T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto ensemble classifier," *Sci. Rep.*, vol. 12, no. 1, p. 15647, Sep. 2022.

[129] S. Aurangzeb, H. Anwar, M. A. Naeem, and M. Aleem, "BigRC-EML: big-data based ransomware classification using ensemble machine learning," *Cluster Comput.*, vol. 25, no. 5, pp. 3405–3422, Oct. 2022.

[130] U. Zahoora, M. Rajarajan, Z. Pan, and A. Khan, "Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier," *Appl. Intell.*, vol. 52, no. 12, pp. 13941–13960, 2022.

[131] G. Y. Kim, J.-Y. Paik, Y. Kim, and E.-S. Cho, "Byte frequency based indicators for crypto-ransomware detection from empirical analysis," *J. Comput. Sci. Technol.*, vol. 37, no. 2, pp. 423–442, Apr. 2022.

[132] J. Du, S. H. Raza, M. Ahmad, I. Alam, S. H. Dar, and M. A. Habib, "Digital forensics as advanced ransomware pre-attack detection algorithm for endpoint data protection," *Secur. Commun. Netw.*, vol. 2022, pp. 1–16, Jul. 2022.

[133] M. Rhode, P. Burnap, and A. Wedgbury, "Real-time malware process detection and automated process killing," *Secur. Commun. Netw.*, vol. 2021, pp. 1–23, Dec. 2021.

[134] W. K. Wong, F. H. Juwono, and C. Apriono, "Vision-based malware detection: A transfer learning approach using optimal ECOC-SVM configuration," *IEEE Access*, vol. 9, pp. 159262–159270, 2021.

[135] M. N. Olaimat, M. Aizaini Maarof, and B. A. S. Al-rimy, "Ransomware anti-analysis and evasion techniques: A survey and research directions," in *Proc. 3rd Int. Cyber Resilience Conf. (CRC)*, Jan. 2021, pp. 1–6.

[136] P. Yosifovich, D. A. Solomon, and A. Ionescu, *Windows Internals, Part 1: System Architecture, Processes, Threads, Memory Management, and More*. Redmond, WA, USA: Microsoft Press, 2017.

[137] F. De Gaspari, D. Hitaj, G. Pagnotta, L. De Carli, and L. V. Mancini, "The naked sun: Malicious cooperation between benign-looking processes," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Springer, 2020, pp. 254–274. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-57878-7

[138] SI Project. (Aug. 2015). *Stratosphere Laboratory Datasets*. [Online]. Available: https://www.stratosphereips.org/datasets-overview

[139] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[140] N. Moustafa. (2015). *The UNSW-Nb15 Dataset*. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[141] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," 2018, *arXiv:1802.10135*.

[142] Microsoft. (Feb. 2015). *Microsoft Malware Classification Challenge (Big 2015)*. [Online]. Available: https://www.kaggle.com/c/malware-classification

[143] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," 2016, *arXiv:1609.03020*.

[144] IC London. (2016). *Resilient Information Systems Security*. [Online]. Available: https://rissgroup.org/ransomware-dataset/

[145] G. Severi, T. Leek, and B. Dolan-Gavitt, "MALREC: Compact full-trace malware recording for retrospective deep analysis," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*. Springer, 2018, pp. 3–23. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-93411-2

[146] ML Laboratory, G Tech, and NYU. (2016). *The MALREC Dataset*. [Online]. Available: https://giantpanda.gtisc.gatech.edu/malrec/dataset/

[147] H. S. Anderson and P. Roth, "EMBER: An open dataset for training static PE malware machine learning models," 2018, *arXiv:1804.04637*.

[148] (2017). *Elastic Malware Benchmark for Empowering Researchers*. [Online]. Available: https://github.com/elastic/ember

[149] M. Nunes. (2018). *Dynamic Malware Analysis Kernel and User-Level Calls*. [Online]. Available: https://doi.org/10.5281/zenodo.1203289

[150] R. Ko, E. Tsen, and S. Slapnicar. (2004). *Dataset of Data Breaches and Ransomware Attacks Over 15 Years From 2004*. [Online]. Available: https://doi.org/10.14264/dfe5027

[151] B. Jethva, I. Traoré, A. Ghaleb, K. Ganame, and S. Ahmed, "Multilayer ransomware detection using grouped registry key operations, file entropy and file signature monitoring," *J. Comput. Secur.*, vol. 28, no. 3, pp. 337–373, Apr. 2020.

[152] (2020). *Botnet and Ransomware Detection Datasets*. [Online]. Available: https://www.uvic.ca/ecs/ece/isot/datasets/botnet-ransomware/index.php

[153] R. Harang and E. M. Rudd, "SOREL-20M: A large scale benchmark dataset for malicious PE detection," 2020, *arXiv:2012.07634*.

[154] Sophos. (Jan. 27, 2020). *Sorel-20m: Sophos-reversinglabs 20 Million Dataset*. [Online]. Available: https://github.com/sophos/SOREL-20M

[155] UCI Machine Learning Repository. (2020). *Bitcoin Heist Ransomware Address Dataset*. [Online]. Available: https://doi.org/10.24432/C5BG8V

[156] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, "BODMAS: An open dataset for learning based temporal analysis of PE malware," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2021, pp. 78–84.

[157] (2021). *Bodmas Malware Dataset*. [Online]. Available: https://whyis young.github.io/BODMAS/

[158] E. Berrueta. (2021). *Ransomware and User Samples for Training and Validating Ml Models*. [Online]. Available: http://dx.doi.org/10.17632/yhg5wk39kf.2

[159] S. R. Davies, R. Macfarlane, and W. J. Buchanan, "NapierOne: A modern mixed file data set alternative to Govdocs1," *Forensic Sci. International: Digit. Invest.*, vol. 40, Mar. 2022, Art. no. 301330.

[160] (2022). *Napierone*. [Online]. Available: http://napierone.com.s3.eu-north-1.amazonaws.com/NapierOne/index.html#NapierOne/

[161] M. Hirano, R. Hodota, and R. Kobayashi, "RanSAP: An open dataset of ransomware storage access patterns for training machine learning models," *Forensic Sci. International: Digit. Invest.*, vol. 40, Mar. 2022, Art. no. 301314.

[162] (2022). *Ransap: An Open Dataset of Ransomware Storage Access Patterns*. [Online]. Available: https://github.com/manabu-hirano/RanSAP/

[163] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Open repository for the evaluation of ransomware detection tools," *IEEE Access*, vol. 8, pp. 65658–65669, 2020, doi: 10.1109/ACCESS.2020.2984187.

[164] C. Grajeda, F. Breitinger, and I. Baggili, "Availability of datasets for digital forensics–and what is missing," *Digit. Invest.*, vol. 22, pp. S94–S105, Aug. 2017.

[165] S. Abt and H. Baier, "Are we missing labels? A study of the availability of ground-truth in network security research," in *Proc. 3rd Int. Workshop Building Anal. Datasets Gathering Exper. Returns Secur. (BADGERS)*, Sep. 2014, pp. 40–55.

[166] 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.

[167] A. Namavar 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.

[168] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1249, 2018.

[169] 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.

[170] A. A. Darem, F. A. Ghaleb, A. A. Al-Hashmi, J. H. Abawajy, S. M. Alanazi, and A. Y. Al-Rezami, "An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning," *IEEE Access*, vol. 9, pp. 97180–97196, 2021.

[171] C. Thapa, K. K. Karmakar, A. H. Celdran, S. Camtepe, V. Varadharajan, and S. Nepal, "FedDICE: A ransomware spread detection in a distributed integrated clinical environment using federated learning and SDN based mitigation," in *Proc. Int. Conf. Heterogeneous Netw. Qual., Rel., Secur. Robustness*. Springer, 2021, pp. 3–24. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-91424-0?page=1#toc

[172] 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.

[173] D. Min, Y. Ko, R. Walker, J. Lee, and Y. Kim, "A content-based ransomware detection and backup solid-state drive for ransomware defense," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 7, pp. 2038–2051, Jul. 2022.

[174] S. Mittal, P. Rajput, and S. Subramoney, "A survey of deep learning on CPUs: Opportunities and co-optimizations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5095–5115, Oct. 2022.
[175] B. A. AlAhmadi and I. Martinovic, "MalClassifier: Malware family classification using network flow sequence behaviour," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, May 2018, pp. 1–13.
[176] Y. Liu, Y. Li, and D. Xie, "Implications of imbalanced datasets for empirical ROC-AUC estimation in binary classification tasks," *J. Stat. Comput. Simul.*, vol. 94, no. 1, pp. 183–203, Jan. 2024.
[177] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," *BioData Mining*, vol. 16, no. 1, pp. 1–23, Feb. 2023.
[178] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.
[179] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102923.
[180] C. Yang, J. Xu, S. Liang, Y. Wu, Y. Wen, B. Zhang, and D. Meng, "DeepMal: maliciousness-preserving adversarial instruction learning against static malware detection," *Cybersecurity*, vol. 4, no. 1, pp. 1–14, Dec. 2021.

JAMIL ISPAHANY (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the School of Computing, Mathematics, and Engineering, Charles Sturt University, Australia. He has been awarded the Cyber Security Cooperative Research Centre (CSCRC) Scholarship. His scholarly pursuits focus on cyber security, machine learning, and malware detection. He has accumulated two decades of professional experience in information technology and cybersecurity, with extensive experience across diverse sectors, including finance, utilities, government, and retail.

MD. RAFIQUL ISLAM (Senior Member, IEEE) is currently an Associate Professor with the School of Computing, Mathematics and Engineering, Faculty of Business, Justice and Behavioral Sciences, Charles Sturt University, Australia. He has a strong research background in cybersecurity with a specific focus on malware analysis and classification, authentication, security in the cloud, privacy in social media, and the Internet of Things (IoT). He is leading the Cybersecurity Research Team and has developed a strong background in leadership, sustainability, and collaborative research. He has a strong publication record and has published more than 180 peer-reviewed research papers, book chapters, and books. His contribution is recognized both nationally and internationally by achieving various rewards, such as professional excellence, research excellence, and leadership awards.

MD. ZAHIDUL ISLAM is currently a Professor of computer science with the School of Computing, Mathematics and Engineering, Charles Sturt University, Australia. His main research interests include data mining, knowledge discovery, privacy-preserving data mining, and applications of data mining/machine learning in various areas, including cyber security. URL: http://csusap.csu.edu.au/zislam/

M. ARIF KHAN (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology Lahore, Pakistan, the M.S. degree in electronic engineering from the GIK Institute of Engineering Sciences and Technology, Pakistan, and the Ph.D. degree in electronic engineering from Macquarie University, Sydney, Australia. He is currently a Senior Lecturer with the School of Computing, Mathematics and Engineering, Charles Sturt University, Australia. His research interests include future wireless communication technologies, smart cities, massive MIMO systems, and cyber security. He was a recipient of the Prestigious International Macquarie University Research Scholarship (iMURS) and the ICT CSIRO scholarships for the Ph.D. degree. He has the competitive GIK Scholarship for the master's degree.