Trends and Future Directions in Automated Ransomware Detection

[1,2*]Abayomi Jegede, [3]Ayotunde Fadele, [4]Monday Onoja, [5]Gilbert Aimufua and [6]Ismaila Jesse Mazadu

[1]Department of Computer Science, University of Jos, Nigeria
[2]Africa Centre of Excellence on Technology Enhanced Learning, National Open University of Nigeria, Abuja, Nigeria
[3]Department of Computer Science, Federal College of Education Zaria, Nigeria
[4]Department of Mathematics and Computer Science, Federal University of Health Sciences, Otukpo, Nigeria
[5]Department of Computer Science, Nasarawa State University, Keffi
[6]Department of Computer Science, Federal University, Wukari, Nigeria

email: [1,2*]jegedea@unijos.edu.ng, [3]ayotundefadele@yahoo.com, [4]mondiono@gmail.com, [5]aimufuagio@yahoo.com, [6]mazadujesse@gmail.com

**Corresponding author*

Received: 21 August 2022 | Accepted: 24 October 2022 | Early access: 28 October 2022

Abstract - *Ransomware attacks constitute major security threats to personal and corporate data and information. A successful ransomware attack results in significant security and privacy violations with attendant financial losses and reputational damages to owners of computer-based resources. This makes it imperative for accurate, timely and reliable detection of ransomware. Several techniques have been proposed for ransomware detection and each technique has its strengths and limitations. The aim of this paper is to discuss the current trends and future directions in automated ransomware detection. The paper provides a background discussion on ransomware as well as historical background and chronology of ransomware attacks. It also provides a detailed and critical review of recent approaches to ransomware detection, prevention, mitigation and recovery. A major strength of the paper is the presentation of the chronology of ransomware attacks from its inception in 1989 to the latest attacks occurring in 2021. Another strength of the study is that a large proportion of the studies reviewed were published between 2015 and 2022. This provides readers with an up-to-date knowledge of the state-of-the-art in ransomware detection. It also provides insights into advances in strategies for preventing, mitigating and recovering from ransomware attacks. Overall, this paper presents researchers with open issues and possible research problems in ransomware detection, prevention, mitigation and recovery.*

Keywords: machine learning, deep learning, neural network, security, ransomware attack, ransomware detection

1 Introduction

Ransomware is malware that hijacks data or systems and prevents legitimate owners of such data or systems from accessing them. Ransomware may encrypt data or lock the system using processes, tools and techniques which make the locking or encryption difficult for a computer expert to reverse. It may also steal sensitive data from victims' computers and networks. Ransomware targets personal computers, business systems (including their data and applications) and industrial control systems. It also attacks internet of things (IoT) spectrum sensors (Celdrán et al., 2022). A ransomware attack uses private key encryption to deny a legitimate user access to his system or data until he pays a ransom (money), usually in bitcoin (Richardson & North, 2017). Ransomware attacks may also involve data exfiltration, whereby attackers copy sensitive files from compromised devices with a threat to revel such files to the public if the owner fails to pay ransom. The malware spreads through email attachments, malicious advertisements and by clicking a link to a malicious website. It locates the drives on the victim's system or network and encrypts the files in each drive to deny the legitimate owners' access to such files (Morhurle &

Patil, 2017). The attacker also provides a file, (or files) which contains instructions for paying the ransom. The decryption key is made available to the victim once the attacker confirms the payment of the ransom. Files infected or encrypted by ransomware usually contain extensions such as .aaa, .micro, .encrypted, .ttt, .xyz, .zzz, .locky, .crypt, .cryptolocker, .vault, or .petya. The extension of each file determines the type of ransomware that infected the file. Examples of ransomware are Reveton, CryptoLocker, CryptoLocker.F and TorrentLocker, CryptoWall, CryptoTear, Fusob and WannaCry (Andronio et al., 2017). Ransomware can be grouped into (1) crypto ransomware, (2) locker ransomware and (3) scareware (Andronio et al., 2017). Figure 1 illustrates the operations of policing (locker) ransomware and encrypting (crypto) ransomware (F-Secure Labs, 2013).

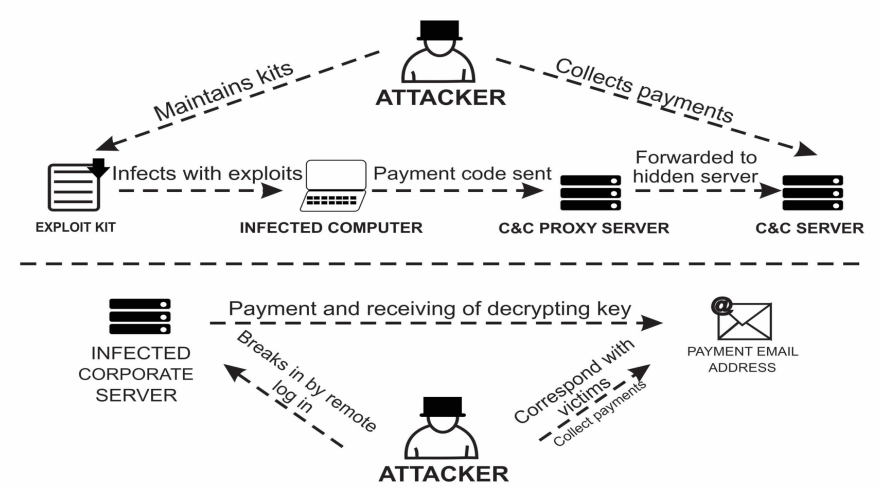

Figure 1: Encrypting ransomware vs. police ransomware operation flowchart

Crypto ransomware is the most common ransomware which attacks computer systems and networks. This category of ransomware uses symmetric and/or asymmetric cryptographic algorithm to encrypt files and data. Crypto ransomware renders encrypted data inaccessible even if the malicious software is removed from an infected device or a compromised storage media is inserted into another device. The infected device can still function and could be used to pay the ransom because the malware does not usually affect critical system files (Savage et al., 2015). Locker ransomware, on the other hand, locks a computer or any other device and prevents the owner from using it (Savage et al., 2015). Locker ransomware affects only the device, without rendering stored data inaccessible. There is also no alteration to the data after the removal of the malicious software. The data can often be recovered by inserting the infected storage device, such as a hard drive, into another system. This makes locker ransomware unattractive for extorting money from victims of attack. A scareware exploits its victims by displaying a warning on their computer screens that the systems have been infected and with a claim that a fake antivirus advertised by the attacker could be used to remove the ransomware (Brewer, 2016). The repeated display of the scareware alert prompts many innocent users to purchase and install the bogus antivirus. Other categories of ransomware include human-operated ransomware (Microsoft Ignite, 2022) and fileless ransomware (Crowdstrike, 2022a). Cyber criminals also use human-operated ransomware to penetrate networks or cloud infrastructure, perform privilege escalation and launch attacks against critical data. It is an active attack which targets an entire organization instead of a single system. Attackers usually leverage on incorrect security configurations to penetrate an entire IT infrastructure, perform lateral movement and exploit vulnerabilities. This results in unauthorized access to credentials of privileged users with the ultimate goal of launching ransomware attacks against IT infrastructures which support critical business operations. Fileless ransomware, on the other hand, uses native and legitimate system tools to launch attacks (Crowdstrike, 2022b). They are difficult to detect because the attack does not require the installation of any code on a victim's system. Hence, anti-ransomware tools do not find any suspicious file to track during an attack. Human-operated ransomware and fileless ransomware may be used to carry out file encryption, locking or data leak depending on the motive of an attacker.

Ransomware poses serious threats to files and devices used by businesses and individuals. It prevents innocent victims from accessing infected files or compromised devices until they pay ransom usually in the form of bitcoin. In many cases, hackers do not provide the decryption key even after a victim pays a ransom. At other times, an attempt to decrypt files using the key provided by an attacker causes further harm to files stored on the system. Technological innovations such as ransomware development kits, ransomware-as-a-service and bitcoins facilitate the persistent increase in ransomware attacks against personal computers, networks and mobile devices (Zetter,

2015). Businesses and individuals suffer losses to the tune of hundreds of millions of dollars annually due to ransomware attacks (Fitzpatrick et al., 2016). The huge amount of money which hackers make from ransomware attacks fuels the frequent development of new versions of the malware. In fact, multiple versions of ransomware have emerged each year since 2013. The evolution of different variants of ransomware which cannot be detected by conventional antivirus and other intrusion detection systems, as well as the huge losses which ransomware attacks inflict on individuals and businesses, highlight the need for innovative, efficient and reliable techniques for effective detection, prevention and mitigation of ransomware attacks.

The paper is novel in the following areas. Firstly, it presents a much more detailed and comprehensive history and chronology of ransomware than other related studies. A related work (Vehabovic et al., 2022) presents the history of ransomware from 2012 to 2021, while our work covers ransomware's history from its inception in 1989 to the latest attacks in 2021. The other study also presents high-level classification of existing ransomware detection methods into four broad categories, with few papers (about forty-seven) reviewed for all the categories, while our paper surveyed almost twice this number and provides much more detailed review of each paper. A significant number of the papers surveyed were published in 2022, unlike the other study which reviewed only a single 2022 paper. Secondly, our paper has a broader scope than the work of McIntosh et al. (2021), which focused primarily on ransomware mitigation, and Oz et al. (2021), whose focus is only on defence/prevention. Our work covers history, detection, defence/prevention, mitigation and recovery. Also, our paper provides an up-to-date review of ransomware attacks by surveying several 2022 papers, while almost all the papers reviewed in McIntosh et al. (2022), and Oz et al. (2021), were published before 2021. Finally, the focus of Dargahi et al. (2019) is completely different from that of our work. The paper presents a taxonomy of crypto-ransomware features using cyber-kill-chain, while the emphasis of our research is on history, detection, defence/prevention, mitigation and recovery. The rest of our paper is divided into the following sections. Section 2 presents the methodology used for the study, while Section 3 covers the historical background and chronology of ransomware attacks. Section 4 discusses the state-of the-art in ransomware detection, while Section 5 is a review of some methods for preventing, mitigating and recovering from ransomware attacks. Section 6 presents suggestions for future research, while Section 7 is the conclusion of the study.

Stages in Ransomware Attack

Ransomware attack involves a number of phases. Figure 2 illustrates the flow of activities required to carry out such an attack.

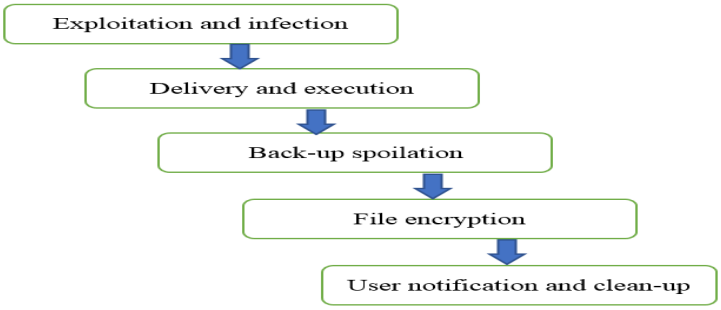

Figure 2: Phases of ransomware attack

An attacker uses exploitation and infection phase to identify vulnerabilities that can be used to launch an attack against a victim computer. The attacker may use a malicious email attachment or an exploit kit for this purpose. For example, the cryptolocker ransomware uses the Angler exploit kit to access and execute on victims' computers. The Angler exploit kit can exploit common vulnerabilities in Adobe Flash and Internet Explorer. The delivery and execution stage involves the installation and execution of the actual ransomware code on the victim's system once there are known vulnerabilities that can support the execution of the malicious payload. Once the file malicious payload executes, it establishes connection with the attacker via the command-and-control mechanism and continues to do further damage. Back-up spoliation involves identification and removal of the system's back-up files and folders to prevent restoration of infected files from back-up. This takes place few seconds after the execution of the ransomware. This is to ensure that victims cannot retrieve compromised files without paying ransom. For example, CryptoLocker and Locky uses vssadmin tool to execute a command that deletes the volume shadow copies from Windows systems. Other variants of these ransomware can identify and delete files from backup folders in order to make recovery a herculean task. File encryption occurs after the removal of backup folders. The process involves a secure key exchange with the command-and-control server to generate encryption keys that will be used to lock the files on the local system. Most modern ransomware variants use strong encryption algorithms such as AES 256 or RSA 1024 which makes it difficult for victims to decrypt infected files.

Ransomware variants such as SamSam performs file encryption locally (on victim systems) without any need to access a command-and-control server via the Internet. Finally, the hacker notifies the victim of the attack and presents instructions for payment of ransom. This occurs after the removal of the back-up files and encryption of the main files. The victim is often asked to pay a ransom within a few days and failure to do so results in an increase in the amount charged for ransom. The payment instructions are usually stored on the hard drive or in the folders containing infected files. At other times, they are saved in specific locations on the hard disk. The malicious executable file automatically deletes itself from the infected system to avoid recovery of useful forensic evidence that would reconstruct the attack and protect against the malware.

2 Methodology

The achievement of the overall objectives of the paper involved the following phases: data collection/information gathering, data extraction/analysis, information synthesis and reporting. Figure 3 is the research process flow, which illustrates the flow of activities involving the phases and the relationship between them.

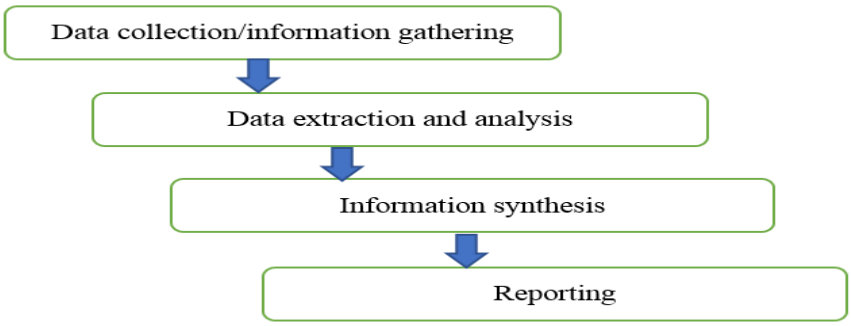

Figure 3: Research process flow

Data collection was performed by selecting relevant and up-to-date journal and conference papers from reputable databases such as IEEE, Springer, MDPI, Elsevier, IET and Archive.org. Other sources include university-based journals, thesis/dissertations and blogs published by reputable organizations such as Microsoft, Crowdstrike, Symantec and Techspot. The materials are then grouped into two main categories, namely, non-technical sources and technical sources. Non-technical sources include materials containing general information on ransomware and as such, provide reliable information for writing the sections on introduction and ransomware history/chronology of attacks. Technical papers that proposed solutions for ransomware attacks are divided into four groups: detection, prevention, mitigation and recovery. A paper is placed in a group depending on the nature or purpose of solution it proposes. Papers that focus on detection are further subdivided into artificial intelligence (AI)-based methods and non-artificial intelligence-based approaches. AI-based approaches are then classified into machine learning methods, deep learning approaches and artificial neural networks approaches, while papers which used non-AI approaches are grouped into packet and traffic analysis categories. Data extraction involved a detailed analysis and summary of each technical paper by identifying the problem the paper addressed, its objective(s), the method/technique used, achievements of the paper in terms of the results obtained, and limitations of the study. Information synthesis was applied to identify similarities or relationships among papers in each group and, if and how a study improved upon, or addressed the limitations of another work. The reporting phase placed papers which addressed similar problems or used similar techniques in the same group, and presented their reviews in the same paragraph. This provides a good flow of communication and enhances the readability of the paper. It also provides readers with a clear understanding of the concepts discussed in the study.

3 Historical Background and Chronology of Ransomware Attacks

Ransomware was first developed in 1989, when Dr. Joseph Popp created a malware called PC Cyborg or AIDS trojan. The malware attacked systems by hiding all folders and encrypting files on the hard disk. The ransomware spread via floppy disks and attackers used a script to request victims to send $189 to a post office box in Panama in favour of PC Cyborg Corporation [6]. The infection prevented users from accessing their computers until ransom was paid and attacks were reversed. The development of strong encryption algorithms has led to the emergence of many variants of the AIDS trojan, which makes it difficult for victims to recover encrypted files without paying ransom. The worst ransomware attack occurred in 2017 with the emergence of the WannaCry Ransomware. This malware encrypts files or systems, and denies legitimate users' access to files or entire devices. A victim can access his files or system only after a ransom is paid and the attacker releases a decryption key. The Wannacry ransomware affected more than 2 million victims cutting across health, business, education and

government sectors. WannaCry encrypts user data and leaves only two files consisting of the encrypted file and a file containing instructions for payment of ransom. The second file also contains a threat that hijacked data will be deleted if the victim fails to pay ransom. The ransomware opens an original file, reads its contents, creates the encrypted version and closes the file (Scaife et al., 2016). India suffered the worst WannaCry ransomware attack with Madhya Pradesh, Maharashtra and Delhi recording 32.63%, 18.84% and 8.76% of total attacks on the country respectively (eScan, 2017). High net worth corporations like FedEx, Nissan, railway companies in Germany, Russian Railways, Megafor Telefonica were also not spared. Many NHS organisations in United Kingdom were severely hit. The attack also caused serious damages to computers belonging to universities and students in China. Well-known internet service providers like RailTel and Vodafone were the most severely affected (Mohurle & Patil, 2017).

Table 1 presents a chronology of major ransomware attacks. The table provides important information on ransomware evolution based on the year a ransomware emerged, the name of the ransomware, its mode of attack, how it spreads, encryption strategy and method used by victims to pay ra`nsom.

Table 1: Chronology of major ransomware attacks

Year	**Ransomware Name**	**Attack mode**	**Mode of spread**	**encryption strategy**	**Ransom payment method**
1989	AIDS Trojan	Encryption of file names	Infected floppy disk	Symmetric encryption	$189 postal order
2005	Trojan PGPcoder	File encryption	Spam email attachment	Asymmetric RSA-1024 encryption	N/A
2006	Trojan Cryzip	Creates password-protected archives of infected files	Spam email attachment	Password locking	No payment; malware code includes password
	Archievus	Encryption of **My Documents** folder	Phishing emails	Asymmetric RSA-1024 encryption	Purchase of 30-digit recovery password
2007	Locker	Display of pornographic image on the machine	Phishing attack	AES and RSA	SMS text message or calling a premium-rate phone number
2008	GPcode.AK	File encryption of subdirectory	Email phishing	Asymmetric RSA-1024 encryption	$100 to $200 in e-gold or Liberty Reserve
2011	60,000 new samples	Varying attack modes	Different modes of spread	Varying encryption and locking methods	Anonymous payment services
2012	Reveton	Password stealing	Clicking malicious link	Malicious JavaScript files	Around $300
	Trojan.Randsom.C	Device locking	N/A	N/A	calling a premium-rate phone number to reactivate Windows license
2013	CyptoLocker	File encryption	Gameover ZeuS banking Trojan botnet;	public and private cryptographic keys	Two Bitcoins (or $100), CashU, Ukash,

			malicious email		Paysafecard, and MoneyPak
	Locker	File encryption	Spam campaigns	AES	$150 via Perfect Money or QIWI Visa Virtual Card number
2014	CryptoDefense	File encryption	Spear phishing email	RSA-2048	earned $34,000 in its first month
	CryptoWall	File encryption	Infected USB drive, email, malicious executables, malicious websites	RSA-2048	more than $1,000,000
2015	LockerPin	Device locking	Adult entertainment app	AES	$500
	Linux.Encoder.1	Encryption of data and web applications files	Exploits the flaw in Magento shopping cart software	AES and RSA	Unspecified amount in bitcoin
2016	Petya	File overwriting and full hard disk encryption	MEDoc tax and accounting software	Master boot record (MBR) and file encryption	$300
	KeRanger	File encryption	Infected web link	RSA	1 bitcoin
	Xbot	File encryption and stealing online banking details	SMS messages	N/A	$100
2017	WannaCry	File and device encryption	Unknown	Hybrid (AES and RSA)	$300 in bitcoin
	Bad Rabbit	Device locking	Drive-by-download on infected websites	Locks users' devices when they click on alicious Adobe Flash installer	$280 bitcoin
2018	GandCrab	File encryption	Infected phishing email, Microsoft Office macros, VBScript and ransomware-as-a-service	Installs on a device and encrypts user files when they access infected email	$500-$600
	Katyusha	File encryption	Malware trojan encrypts and adds 'Katyusha' extension to infected files	Infects networks using EternaBlue and DoublePulsar exploits	0.5 bitcoin

	Ryuk	File encryption	Massive spam attacks and exploit kits	Symmetric AES-256 and asymmetric RSA-2048 encryption	15-50 bitcoins
2019	Prolock/ PwndLocker	File lock/encryption	Qakbot Trojan	Asymmetric RSA-2048 encryption	Bitcoin
	LockerGoga	File encryption and file wiping	Logs users out of systems, encrypts files and deactivate devices	Cryptographic encryption and deletion of infected files	N/A
	PewCrypt	File encryption	Spam email messages	Symmetric 256-bit Advanced Encryption Scheme (AES-256)	Free
	Dharma v2019	File encryption	Malicious email	Symmetric AES-256 algorithm	N/A
2020	Nefilim	File encryption	Remote desktop protocol (RDP) attack	AES-256 encryption for victim's files; RSA-2048 algorithm to encrypt the AES-256 keys	Via email communication
	Ransomware Name	Attack mode	Mode of spread	encryption strategy	Ransom payment method
	Paradise v2020	File encryption	Spam message containing internet query attachments	RSA-1024 and RSA-2048 algorithms	No ransom. Tools are available to retrieve encrypted files
	Maze	File encryption	Exploit kits such as Fallout and Spelva	RSA and ChaCha20 stream cipher	$6m - $15m
	REvil	File encryption/file blocking	Phishing email and malicious attachment	AES or Salsa20	$70m in bitcoin
	Tycoon	Password exploitation of file servers and domain controllers	Insecure connection to an RDP server and a malicious (trojanized) Java Runtime Environment	RSA	N/A
	NetWalker	Full Windows device encryption	Network-wide executable files and VBS script attachments in Corona virus phishing emails.	Salsa20	More than $30m total ransom since March 2021
2021	Dark side	File encryption and data exfiltration	VPN password	Lightweight Salsa20 with RSA-1024	75 bitcoin or $4.4m

	ReVil	File encryption/file blocking	Vulnerability in Microsoft Exchange servers	AES or Salsa20	$50m in Monero cryptocurrency demanded
	Phoenix locker	File encryption on desktop and network shares	Spam emails	RSA-2048 algorithm	$40m
	ContiLocker	File encryption and data exfiltration	Via unprotected remote desktop protocol (RDP) port	RSA-4096 and AES-256-CBC	$2.6m
	Avaddon	File encryption, data exfiltration and DDoS	Malicious JavaScript files	AES-256	$40,000 or its equivalent in bitcoin

The table shows that the development of ransomware and deployment of ransomware attacks have been on the rise since 1989 when the first known ransomware emerged. Most ransomware attack involves encryption of files and sub-directories. The devices can still function, but the infected files are inaccessible to legitimate users. A less common form of attack involves blocking users from gaining access to their devices, even if the files stored on such devices are accessible. New variants of malware have also emerged each year since 2013. This is because of the availability of sophisticated tools that enable attackers to easily craft ransomware scripts as well as huge amounts of money hackers make from ransom payment. Maze, REvil, Ryuk, Tycoon and NetWalker are currently the five most dangerous ransomware attacks (Ransomware Attacks, 2021). Several factors enhance the growth of ransomware and persistent increase in ransomware attacks. These include easy procurement of powerful encryption (symmetric and asymmetric) algorithms, which enables attackers to easily craft a ransomware tailored for a specific attack, or environment and availability of effective infection vectors such as spam email and malvertising, which ensure that a ransomware spreads rapidly to as many users as possible (Adamov & Carlson, 2017). Other factors are easy accessibility of victims to cryptocurrency for ransom payments (including the ease with which attackers can convert cryptocurrency to cash without any trace) and the availability of Ransomware as a Service (RaaS) also enables unskilled and less knowledgeable attackers obtain customize ransomware and track victims via a user interface (Gellegos-Segovia et al., 2017). The creators of RaaS earn a percentage of profits from ransomware attacks launched via their platforms.

4 Ransomware Detection

Research show that ransomware attacks are on the rise and have doubled in the first quarter of 2020 due to increase in remote working culture imposed by COVID-19 pandemic. Many individuals who work from home do not practice the same cybersecurity measures commonly imposed in the office environment. Also, most remote workers use personal devices which are not adequately equipped with security mechanisms such as antimalware packages, firewall, intrusion detection/prevention systems, password management tools and encryption software. Ransomware leverages on new vulnerabilities found in systems and networks, using attacks focus on both small, medium and big companies who imbibe the remote working culture. Apart from encrypting files and locking devices, ransomware can also use sophisticated techniques to carry out data exfiltration. This resulting exposure of sensitive information may lead to severe security concerns and privacy violations. This is addition to financial losses and reputation damage suffered by victims. Ransomware attack against a health facility may result in loss of life such as in the case of a Dusseldolf University hospital patient where an attack interrupted emergency services and the hospital management had to send the patient to another hospital 17 miles away (Fingers, 2020). The patient eventually died as a result of delay in treatment. Ransomware payment is also a means by which attackers extort several millions of dollars from innocent victims every year (Symantec Corporation, 2016) Ransomware attacks account for more than 41% of cyber insurance claims in 2020 and it is projected that total losses which have organizations suffer from ransomware attacks may hit $20 billion at the end of 2020 (Potoroaca, 2020). The money which organizations use to pay ransom can be channeled to other productive ventures resulting in the overall growth of the business. These concerns highlight the need for efficient and reliable methods for ransomware detection, prevention, mitigation and recovery. Ransomware detection methods are generally categorized into automated and manual. Automated approaches rely on the use of tools to detect and report ransomware attacks. Such tools are usually software packages which may also possess the ability to block attacks. Manual detection methods focus on regular inspection of files and devices for obvious signs of attacks. This includes checking for changes in file extensions and whether authorized users can access files and devices. That

is, checking whether a malware attack has not modified files and authorized users have not been blocked from accessing their devices and files. The flow of presentation in this section is illustrated in Figure 4.

4.1 Automated Ransomware Detection

Existing approaches for ransomware detection predominantly focus on system level monitoring, for instance, by tracking the file system characteristics. Automated ransomware detection approaches can be divided into two major categories namely, artificial intelligence (AI)-based methods and non-artificial intelligence (non-AI)-based methods. AI-based methods commonly use techniques such as machine learning (ML), deep learning (DL) and artificial neural network (ANN) for ransomware detection. Some tools apply variants of these techniques or a hybrid approach using a combination of two or more techniques to address the menace of ransomware attacks. Non-AI methods use approaches such as packet inspection and traffic analysis to detect ransomware. A major strength of automated approaches is their ability to detect, block and recover from ransomware attack without human intervention. The tools also possess high level accuracy and reliability in terms of ransomware detection, prevention and recovery.

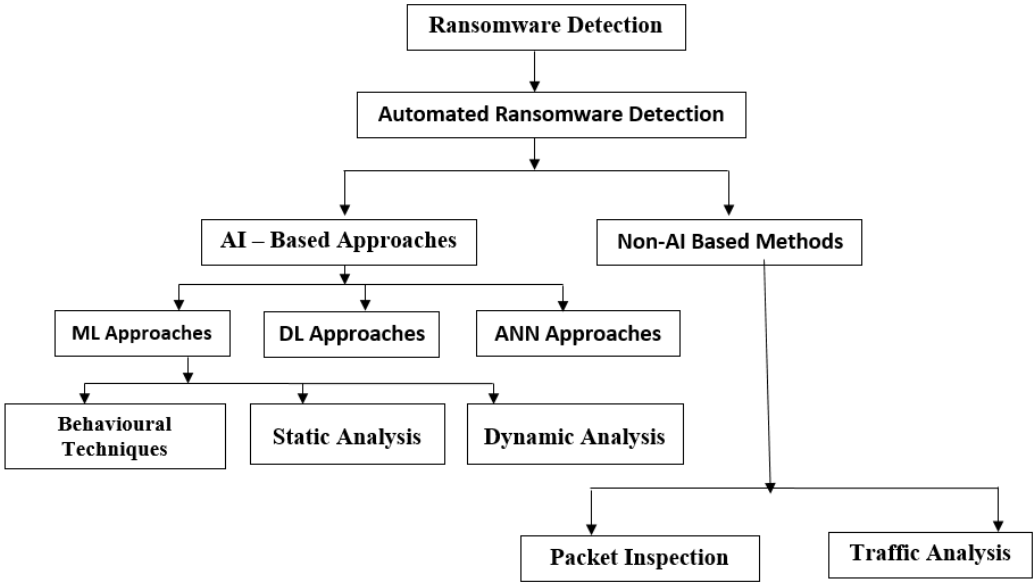

Figure 4: Flow of presentation on ransomware detection

4.1.1 Artificial Intelligence-Based Methods

Artificial intelligence-based methods use machine learning (such as behavioural techniques and static and dynamic analysis), deep learning and artificial neural network to perform automated detection of ransomware attacks.

4.1.1.1. Machine Learning Approaches

Machine learning (ML) is a branch of artificial intelligence which provides systems with the ability to learn from, and detect patterns in existing data, while making decisions with little or no human intervention (Dontov, 2019). It is a method commonly used to automate analytical model building. ML techniques enable computers to make predictions based on patterns found in large datasets. The algorithms are able to adapt to changes and make improvements as the size of the dataset increases. The ability of ML to make predictions based on file behaviour as well as known and unknown datasets makes it a viable tool for detecting previously unknown ransomware variants. However, machine learning techniques require a minimum of between 50 and 1,000 data points to make reliable prediction. Few samples may result in overfitting and biased prediction. Also, training machine learning algorithms require significant amount of time. File behaviour detection is the major application of machine learning to ransomware detection. ML algorithms use specialized analysis (such as interactive debugging or post mortem code execution analysis) to extract large amount of salient and discriminant information in order to learn the behaviour of a legitimate or normal application. ML-based ransomware detection tools perform detailed analysis of legitimate code execution and are able to identify malicious applications. Such tools make intelligent

decisions and prompts specific actions by leveraging on their ability to distinguish between normal and abnormal program execution. The machine learning approaches explored in this study are behavioural techniques as well as static and dynamic analysis.

Behavioural Techniques

A normal application behaviour is measured from both user perspective and resource perspective. A normal behavioural baseline is established based on what represents normal or routine operations of a computer system or network. Such operations may include logins, file access, user and file behaviors, resource utilization, and other important signs of normal activity (Acronis International, 2021). The duration of the learning process depends on the amount of data needed to establish a baseline to represent normal system behaviour. The tool identifies and scrutinizes behavioural anomalies which do not fall within the normal behavioural pattern represented by the baseline. (Juan et al., 2017) proposed a ransomware detection and prevention model for unstructured dataset extracted from Ecuadorian control and regulatory institution (EcuCERT) logs. The approach uses machine learning techniques to detect abnormal behavioral patterns associated with Microsoft Windows-based ransomware. Feature selection was applied to the Log data to extract the most useful and discriminating information that represents a ransomware threat. The extracted information represents the feature set which serves as input for automatic learning algorithms. The algorithms use the input feature set to model abnormal behavioral patterns in order provide timely and reliable detection of ransomware. There was an attempt to address the limitations of signature-based methods in detecting ransomware attacks which evolve daily due to availability of code obfuscation techniques and creation of new polymorphic variants (Shaukat & Ribeiro, 2018). This is necessary because generic malware attack vectors do not adequately capture the specific behavioral patterns of cryptographic ransomware and as such, not sufficient or reliable enough for ransomware detection. The proposed approach known as RansomWall is a layered and hybrid mechanism based on the application of static and dynamic analyses to generate a new set of features that model ransomware behavior. The approach uses a strong trap layer for early detection of ransomware and is suitable for detecting zero-day attacks. An evaluation of RansomWall and Gradient Tree Boosting Algorithm on 574 samples of 12 Microsoft Windows operating system-based cryptographic ransomware produced 98.25% detection rate and very low (almost zero) false positives. It is also able to detect 30 zero-day attack samples, with less than 10% detection rate compared to 60 VirusTotal security engines. CryptoDrop was developed to provide early detection of ransomware based on suspicious file activity (Scaife et al., 2016). It uses a set of behavioral features to terminate any process that alters a large amount of the user's data. CryptoDrop can integrate common ransomware features to support rapid detection with low false positives. Experimental analysis shows that CryptoDrop is an efficient tool for ransomware detection and prevention. It is able to prevent execution of ransomware files with a median loss of only 10 files out of almost 5,100 tested files. Overall, the approach leverages on behavioral analysis to minimize data loss due to ransomware attacks. A limitation of CryptoDrop is its inability to determine the intent of attack indicated by changes in file behaviour. An example is a situation where the tool cannot determine whether a set of documents is encrypted by the user or ransomware. The system simply notifies the user who decides whether a suspicious activity is desirable or not. CryptoDrop flags legitimate activities such as compression whose behavior is normal, expected, desirable, and not actually invasive. It is necessary for future versions to possess the ability to distinguish legitimate bulk transformation activities such as file compression from malicious attacks.

Table 2 presents a summary of previous studies on behavioural techniques for ransomware detection.

Table 2: Summary of related works (behavioural techniques)

Author	Problem addressed	Method used	Result	Limitation
Shaukat & Ribeiro (2018)	Ransomware detection	Layered and hybrid mechanism (RansomWall)	Suitable for detecting zero-day attacks	N/A
Scaife et al. (2016)	Ransomware detection	Evaluation of RansomWall and Gradient Tree Boosting Algorithm (CryptoDrop)	Median loss of only 10 files out of almost 5,100 tested files	Inability to determine the intent of attack indicated by changes in file behavior

Makinde et al. (2019)	To detect the susceptibility of a real network system to ransomware attack	Machine Learning	Correlation above 0.8	It simulated the behaviour of few users
Ahmad et al. (2019)	To distinguish members of the Locky ransomware	Behavioural ransomware detection approach (parallel classifiers)	Highly accurate detection with low false positive rate	N/A
Zahra & Sha (2019)	Detecting Cryptowall ransomware attack	Command and control (C&C) server black listing	Extracts TCP/IP header from web proxy server which serves as the gateway to TCP/IP traffic.	The model was not implemented to demonstrate its accuracy and effectiveness in detecting ransomware and their modes of attack against different operating system environments
Singh et al., (2022)	Detection of previously unknown ransomware families and classification of new ransomware attacks	Examines access privileges in process memory to achieve easy and accurate detection of ransomware	accuracy ranges between 81.38% and 96.28%.	N/A

A variant of behavioural detection approaches used a machine learning baseline model for simulating and predicting the individual network user behaviour pattern at the micro level in order to detect possible scenarios that may indicate a vulnerability or an actual ransomware attack (Makinde et al., 2019). The goal was to detect the susceptibility of a real network system to ransomware attack. A comparative evaluation of the results obtained from the simulated network and the log data obtained from the server in the real-life network system indicates a realistic model with a correlation above 0.8. A limitation of this approach is that it simulated the behaviour of few users. Future works should focus on using tools for big data analytics to simulate the behaviour of a large number of users. A more recent behavioural ransomware detection approach used two parallel classifiers to distinguish members of the Locky ransomware family according to their types (Ahmad et al., 2019). The method focused on early detection based on behavioural analysis of ransomware network traffic in order to prevent a ransomware from connecting to command-and-control servers and executing harmful payloads. The study used a dedicated network to collect network information and extract relevant features of network traffic. The extracted features of the Locky ransomware family are processed by two independent (parallel) classifiers working on data at packet and datagram levels. Experimental results show that the method is able to extract valid features and provides a high level of effectiveness in tracking the activities of ransomware on the network. It also offers highly accurate detection with low false positive rate. Zahra and Sha (2019) proposed a domain-specific framework for detecting Cryptowall ransomware attack based on the communication and behavioral analysis of the ransomware in an IoT environment using command and control (C&C) server black listing to detect ransomware attacks. The method extracts TCP/IP header from web proxy server which serves as the gateway to TCP/IP traffic. It also extracts source and destination IP addresses and compares them with blacklisted IP of Command-and-Control servers. A ransomware is detected if the source or destination IP matches ransomware attack for IoT devices. However, the model was not implemented to demonstrate its accuracy and effectiveness in detecting ransomware and their modes of attack against different operating system environments. A very recent approach to behavioural-based detection leverages on access privileges in process memory to achieve easy and accurate detection of ransomware (Singh et al., 2022). The method can also detect previously unknown ransomware families and classify new ransomware attacks using the access privileges a file or an application possesses and the area of memory it intends to access. The helps to identify the behaviour of an executable, and detect its intent before it causes serious damage to legitimate files and applications. Experimental results based on these multiple algorithms produced good detection accuracy which ranges between 81.38% and 96.28%.

Static and Dynamic Analysis

A novel detection technique based on static analysis extracts features directly from raw ransomware binaries using frequent pattern mining (Khammas, 2020). It also uses Gain Ratio technique to select 1000 features for optimal ransomware detection. Random forest classifier was used to analyze the impact of trees seed numbers on the detection process. Experimental results show that the detection rate of proposed approach is 97.74%. Direct extraction of raw ransomware binaries results in a remarkable increase in the speed of detection. An enhanced approach to ransomware detection integrates dynamic analysis with machine learning (Hwang et al., 2020). It is a hybrid ransomware detection model based on Markov model and Random Forest model. The approach uses Windows API call sequence pattern to build a Markov model which extracts the unique features of ransomware. This is followed by using Random Forest to model the remaining data in order minimize error rates. The two-stage mixed detection technique achieved good detection rates with an overall accuracy of 97.3%, 4.8% FPR (false positive rate) and 1.5% FNR (false negative rate). A similar approach known as *EldeRan* uses dynamic analysis to detect ransomware at run-time (Sgandurra et al., 2016). The technique leverages on the fact that run-time features exhibited by ransomware samples are similar for all ransomware families. *EldeRan* performs dynamic analysis and ransomware detection by monitoring the actions carried out by applications when they are first installed and checking for obvious signs of ransomware. The result of experiments carried out on a dataset of 582 ransomware and 942 goodware applications, shows that the approach achieves an area under the ROC curve of 0.995. A major strength of the *EldeRan* lies in its ability to perform dynamic analysis and ransomware detection even if the entire dataset of a ransomware family is not available. This supports early detection of new ransomware variants.

An improved technique for ransomware detection used an integrated approach, which combines static and dynamic analysis (Bazrafshan et al., 2013). It is an analysis framework based on support vector machines, which uses “run-time” and “static code” features for early detection of known and previously unknown ransomware variants. The results of experiments based on a wide array of ransomware types suggest that the integrated approach provides better ransomware detection than using either static analysis or dynamic analysis individually. The integration of static and dynamic analysis has also been used to analyze ransomware threats against mobile devices and perform mobile ransomware detection (Yang et al., 2015). The proposed approach combines the results of static and dynamic analysis for detecting ransomware threats and attacks against mobile applications. It is a two-phase approach which integrates data states and software execution on the critical test path of the Android API. The first phase is static analysis which detects the likelihood of an attack by using API, existing attack patterns and dynamic analysis to execute a program in a limited and restricted scope and comparing whether the detected path conforms with existing attack patterns. The second phase (which is runtime dynamic analysis) uses dynamic inspection to detect the nature of attack and possible violation of data confidentiality (such as web browser cookie) without compromising sensitive and secured data sources in mobile device. A related work detects unknown ransomware by using the most discriminating API calls to train a classifier (Sheen & Yadav, 2018). The approach was applied on an imbalanced dataset consisting of unequal amounts of ransomware and benign data. Experimental results show that the approach is more suitable for random forest than decision tree or KNN. Random forest produced the best detection rate of over 98% because it is more robust against class imbalance than decision tree and KNN. A limitation of this study is class imbalance in the dataset due to the difference in the number of samples in the ransomware class and benign class. A future work should apply the same technique on a balanced dataset using the same classifiers and observe the outcome. An improved approach integrates feature generation engines and machine learning for analyzing malware samples obtained from raw binaries, assembly codes, libraries, and function calls in order to identify the goal malicious codes intend to achieve. Poudyal et al. (2018) applied different supervised ML techniques on features extracted from ransomware and benign binaries. Performance evaluation results show that the approach has detection accuracy which ranges from 76% to 97% depending on the ML classifier used. Seven out of the eight classifiers achieved a detection rate of at least 90%. The study also revealed better ransomware detection rates when static level analysis is applied to data obtained by integrating ASM-level and DLL-level features. Similarly, Dehghantanha et al. (2018) proposed a Decision Tree (J48) classifier known as NetConverse, for high speed and reliable detection of Windows ransomware. Experimental results based on conversation-based network traffic features dataset show a true positive detection rate of 97.1% using the Decision Tree (J48) classifier. Static and dynamic techniques can also be used for real time detection and prevention of ransomware attack (Lalson et al., 2019). The technique offers a robust and an effective protection against a variety of ransomware. The approach halts attacks before the system or network experiences a significant damage. However, the proposed method cannot perform the recovery of infected files. It is also possible for a ransomware to encrypt some files before it is actually detected or blocked. Lee et al. (2022) addressed the ineffectiveness of static analysis against obfuscating ransomware, which hides their behaviour to evade detection and low-speed detection of dynamic analysis by proposing a statistical analysis which uses heuristics to distinguish between normal files and those attacked by ransomware. The approach

provides real-time detection of known crypto-ransomware variants. It is also efficient with about 13% overhead required during the detection process.

Recent ML approaches such as the one proposed by Rani and Dhavale (2022) used a number of machine learning models such as decision tree, random forest, KNN, SVM, XGBoost and Logistic Regression to build an effective proof of concept for a product specific ransomware. The proposed solution is efficient and reliable with an accuracy of 98.21%. Similarly, three different machine learning algorithms namely decision tree (J48), random forest and radial basis function (RBF) were applied on 1000 dominant features obtained from raw, byte-level ransomware data using the gain ratio feature selection method (Khammas, 2022). The results from experiments show that random forest is the most effective of the threes algorithms with ~ 98% accuracy and the most suitable feature size is 1000 attributes. An enhanced approach integrates ensemble learning with voting-based method, monitors memory usage, system call logs, CPU usage and performs static and dynamic analysis of text, permissions and network-based features (Ahmed et al., 2022). Experimental results based on malicious and benign features (static and dynamic) obtained from Android malware applications show that the proposed technique can detect unknown ransomware attacks based on the behaviour of malicious applications. The technique is also robust against adversarial evasion attacks as demonstrated by its high detection accuracy when tested with 1-bit, 10-bit, 20-bit, 30-bit and 40-bit crafted ransomware data. Talabani and Abdulhadi (2022) proposed two rule-based models to address the low accuracy of ransomware detection tools which use data mining and machine learning techniques. The models known as Partial Decision Tree (PART) and Decision Table were applied to bitcoin dataset consisting of 61,004 samples of 29 ransomware families with ten descriptive and decision attributes. Experimental results show that the PART algorithm provides better performance in terms of accuracy (96.01%), recall (96%), precision (95.9%) and F-Measure (95.6%) than Decision Table. Experimental results show that it is necessary to carry out additional investigation on the application of PART to predictive modelling tasks in ransomware detection experiments.

A summary of previous studies which used static and dynamic analysis for ransomware detection is presented in Table 3.

Table 3: Summary of related works (static and dynamic analysis)

Author	**Problem addressed**	**Method used**	**Result**
Khammas (2020)	Ransomware detection	Random forest technique	Detection rate is 97.74%.
Hwang et al. (2020)	An enhanced approach to ransomware detection.	Markov model and random forest model	Overall accuracy of 97.3%, 4.8% FPR (false positive rate) and 1.5% FNR (false negative rate
Dehghantanha et al. (2018)	High speed and reliable detection of windows ransomware	Netconverse (decision tree (j48) classifier)	True positive detection rate of 97.1%
Rahman & Hasan (2019)	Improved technique for ransomware detection	Analysis framework based on support vector machines	Integrated approach provides better ransomware detection than using either static analysis or dynamic analysis individually.
Jasmin (2019)	Distinguishing ransomware traffic from normal traffic	Random forest, support vector machine and logistic regression algorithms	Random forest has the best detection rate of 99.9% and a false positive rate of 0%.
Ameer (2019)	Ransomware detection	Static and dynamic analysis	Detection and classification accuracy of 100%

Talabani & Abdulhadi (2022)	Low accuracy of ransomware detection tools which use data mining and machine learning techniques	Partial Decision Tree (PART) and Decision Table	accuracy (96.01%), recall (96%), precision (95.9%) and F-Measure (95.6%)

Several enhanced machine learning techniques have been proposed for effective and reliable detection of ransomware. These techniques are meant to address the weaknesses in existing ML-based ransomware detection methods. One of such improvements addressed the limitation of detection techniques (such as sandbox analysis and pipelines) due to their inability to isolate a sample and handle the delay in analyzing isolated ransomware samples (Adamu, 2019). The approach predicts ransomware using a dataset consisting of 30,000 attributes which serve as independent variables. Feature selection was used to obtain five attributes used as input to support vector machine algorithm. The method has promising ransomware detection rate with accuracy of 88.2%. Another improvement focused on detecting ransomware in cloud storage instead of the local system (Matthias, 2018). It is a hybrid technique which integrates 'guilt by association' assumption with content-based, metadata-based and behaviour-based analysis to minimize the false positive rate. This involves the use of file versioning of the cloud storage to delay the recovery and transferring the supervision of the recovery to the end user. The only responsibility of the end-user is to supervise the recovery. Users are provided with classification information which allows them make informed decisions and prevent false positives. The approach provides improved detection accuracy and reliable recovery. A novel approach used network connection information, certificate information and machine learning for network-level ransomware detection (Jasmin, 2019). The method can be used in conjunction with system-level detection to provide early detection of ransomware attacks. The technique extracts and models ransomware features based on three major characteristics of network traffic namely, connection-based, encryption-based, and certificate. It is a feature model which used random forest, support vector machine and logistic regression algorithms to distinguish ransomware traffic from normal traffic. Experimental results based on a variety of datasets showed that random forest has the best detection rate of 99.9% and a false positive rate of 0%. Another enhanced detection approach is a decision tree model based on big data technology, which exploits Argus for packet preprocessing, merging, and labeling malware file (Wan et al., 2018). Biflow was used to replace the packet data and reduce the data size by a factor of 1000 (that is, 1000:1). Feature selection and feature concatenation were employed to extract and combine the characteristics of a complete network traffic. The method used six feature selection algorithms in order to achieve better classification accuracy. A recent and an innovative ransomware detection method used machine learning to monitor power consumption of Android devices (Azmoodeh et al., 2018). The proposed approach distinguishes ransomware from benign applications by monitoring the energy consumption patterns of various Android processes. This is achieved by collecting and analyzing the unique local fingerprint of ransomware's energy consumption. Experimental results show that the method achieved high detection and precision rates of 95.65% and 89.19% respectively. It also has better accuracy, recall rate, precision rate and F-measure than K-Nearest Neighbor, Neural Network, Support Vector Machine and Random Forest. Another enhanced solution is a novel lightweight approach known as RanDroid for automated detection of polymorphic ransomware (Alzahrani et al., 2018). The technique detects new ransomware variants on Android platforms using the structural similarity measures between features extracted from an application and a set of threat data extracted from known ransomware variants. The similarity measures used are Image Similarity Measurement (ISM) and String Similarity Measurement (SSM). Further information was extracted by applying linguistic analysis on the app's code behavioural features and image textural strings. The approach addressed the limitations of static analysis by performing dynamic and static analyses in order to mitigate ransomware attacks without modifying the Android OS and its underlying security module. An evaluation of RanDroid based on 950 ransomware samples showed that the approach can detect ransomware based on evasive techniques such as sophisticated codes or dynamic payloads. A related work proposed a hybrid solution based on the integration of static and dynamic analysis for detecting Android ransomware and distinguishing ransomware from other malware (Ameer, 2019). The approach applied static analysis on permissions, text, and network-based features. It also applied dynamic analysis on the memory usage, system call logs, and CPU usage. The results of experiments based on features extracted from ransomware and benign samples show that technique can mitigate evasive ransomware attacks. It is also able to detect and classify unknown ransomware with 100% accuracy.

4.1.1.2 Deep Learning Approaches

Deep learning techniques are aimed at addressing the shortcomings of conventional supervised ransomware detection tools. The goal is to enhance the accuracy and reliability of results obtained from a ransomware detection activity. Deep learning techniques perform automatic feature generation and are very suitable for unstructured datasets. The techniques also require very little or no human intervention (good self-learning capabilities). Deep learning algorithms are very suitable for classifying audio, text and image data. This enhances their effectiveness

at detecting textual and image ransomware data. However, training deep learning algorithms requires a very large amount of data. This makes the algorithms unsuitable for general purpose applications especially those requiring small data points or sizes. Other limitations of deep learning include the need for high processing (CPU) power and inability to easily adapt to real life datasets. A recent application of this approach is a deep learning based semi-supervised framework, which extracts inherent, unlabeled and previously unknown features of new ransomware variants (Sharmeen et al., 2020). The framework also provides an adaptive detection model by integrating the unsupervised learned model with supervised classification. Experimental results based on real ransomware data with a dynamic analysis testbed shows that the method is highly accurate at detecting different kinds of ransomware compared to existing supervised approaches. Another deep learning approach for automated behavioural-based ransomware detection applied dynamic analysis on data obtained from Application Programming Interface (API) calls made by the executable (Maniath et al., 2017). This approach uses a word sequence to represent the list of API calls made by an executable file. It applied Long-Short Term Memory (LSTM) networks for binary sequence classification of application programming interface (API) calls a suitable method for detecting ransomware behavior. The approach detects ransomware behaviour using API calls obtained from systems logs of modified sandbox environment. It is a suitable method for reliable analysis and detection of large malwares samples. A related study proposed a deep learning technique based on features extracted from permissions and API calls for detecting Android ransomware (Wongsupa, 2018). AndroGuard (a python library) was used for feature extraction, while the ransomware detection framework was implemented on Keras, using multilayer perceptron (MLP) with back-propagation and supervised learning algorithm. The results of experiments on real-world applications show that the accuracy is 98% for MLPs with more than 3 hidden layers and moderately sized neurons. However, the use of MLPs with two hidden layers and large number of neurons results in low detection accuracy of between 45% and 60%. A novel deep learning approach to ransomware detection extracts salient behavioral features from labeled ransomware data (Aragom et al., 2016). It is a novel architecture which combines deep packet inspection with machine learning. The model can detect and prevent various types cryptographic ransomware. Experimental results show that the deep learning model achieved a detection accuracy of 93.92%, which makes it suitable for timely detection of unknown ransomware in high-speed network. Table 4 is the summary of related works which used deep learning techniques to implement automated ransomware detection systems.

Table 4: Summary of related works (deep learning approaches)

Author	**Problem addressed**	**Method used**	**Result**	**Limitation**
Sharmeen et al. (2020)	To enhance the accuracy and reliability of results obtained from a ransomware detection activity	Deep learning based semi-supervised framework	Highly accurate at detecting different kinds of ransomware compared to existing supervised approaches	N/A
Maniath et al. (2017)	Automated behavioural-based ransomware detection	Deep learning techniques	The approach detects ransomware behaviour using API calls obtained from systems logs of modified sandbox environment	N/A
Wongsupa (2018)	Detecting Android ransomware.	Deep learning technique and Supervised learning algorithm	The results of experiments on real-world applications show that the accuracy is 98% for mlps with more than 3 hidden layers and moderately sized neurons	The use of mlps with 2 hidden layers and large number of neurons results in low detection accuracy of between 45% and 60%.
Aragom et al. (2016)	Detection and prevention of various types cryptographic ransomware.	Combines deep packet inspection with machine learning	Detection accuracy of 93.92%,	N/A

Vinayakumar et al. (2017)	Effective detection and classification of ransomware	Enhanced deep learning technique	Classification accuracy of 0.98 (98%)	The experimental results do not represent the actual situation involving more complex architecture settings
Olani et al. (2022)	Detection of ransomware by monitoring and analyzing changes in the distribution hardware performance counter data.	Deep learning	Classification accuracy of 98.6% and recall score of 84.41%.	N/A

An enhanced deep learning technique applied shallow and deep networks on features extracted from API calls for effective detection and classification of ransomware (Vinayakumar et al., 2017). The study explored a number of network parameters and structures to obtain the best architecture for the multi-layer perceptron (MLP). This involved up to 500 epochs with a learning rate between 0.01 and 0.5. The results of various experiments showed that MLP has very high accuracy of 1.0 (100%) in distinguishing ransomware from benign samples. It was also able to classify ransomware into their families with an accuracy of 0.98 (98%). This shows that the approach can detect and classify ransomware better than other classical machine learning techniques. However, the proposed approach is a very simple MLP network, which does not impose high computational burden on hardware and monolithic training environment. The experimental results do not represent the actual situation involving more complex architecture settings. A future work should focus on using more complex MLP network to perform the same experiments on state-of-the-art hardware in a distributed environment. A recent a deep learning model monitors changes in the distribution hardware performance counter data across the system and analyzes relevant information to achieve effective and efficient detection of ransomware (Olani et al., 2022). The information extracted is specifically related to events which indicate behaviour that distinguishes a ransomware from a benign application. The results of experiment based on different ransomware families show that the model is effective with ransomware classification accuracy of 98.6% and recall score of 84.41%. The model is also effective for detecting zero-day attack as demonstrated by experiments based on previously unknown CoronaVirus, Ryuk, and Dharma ransomware variants.

4.1.1.3 Artificial Neural Network Approaches

Neural networks have wide applications which makes them suitable for detecting different types of ransomware data (text or image) and ransomware variants. The ability of neural networks to perform continuous learning makes them suitable for adapting to new ransomware data and detecting zero-day ransomware attacks. However, neural network techniques are hardware dependent and susceptible to data dependency. They also deny human analysts from tracking data processing tasks and checking for deviations (black box nature). Agrawal (2019) proposed an enhanced technique which leverages on the ability of recurrent neural networks to establish a relationship among events which follow a particular sequence. The technique known as Attended Recent Inputs-Long Short-Term Memory (ARI-LSTM) uses attention mechanisms to extract the pattern of events created by ransomware sequences. The approach leverages on the ability of recurrent neural networks to provide high detection accuracy for sequence learning models. An LSTM is a type of recurrent neural network which possesses the ability to establish a relationship among a sequence of events caused by ransomware attack (Shmidhuber & Cummins, 1997; Gers, 2000). ARI enhances neural cells by incorporating attention in learning from ransomware sequences. It uses the concept of a subsequence to extract local event patterns in ransomware sequences to learn from a recent history of ransomware. An evaluation of ARI-LSTM using ransomware and benign executables captured from Windows operating system showed that the technique has better detection rate than LSTM. A much finer scale evaluation of detection accuracy showed that the technique has a False Positive Rate (FPR) set of 2%. Generally, ARI-LSTM possesses much better performance accuracy (or detection rate) of 91% with low values of FPR thus establishing the potency and efficiency of attention mechanisms in learning local patterns. Similarly, identification of important and unique features of ransomware can be used to detect an attack (Arslan, 2020). This is achieved by using transfer learning based deep convolutional neural networks to perform feature engineering in order to analyze important properties and behaviors of a ransomware. The technique leverages on the ability of

neural networks to detect some attributes, states, and patterns of ransomware files. Feature engineering and analysis were performed on static and dynamic datasets consisting of 3646 samples (1700 Ransomware and 1946 Goodware) and 3444 (1455 Ransomware and 1989 Goodware) samples respectively. Experimental results show that relevant features for ransomware detection are registry changes, application programming interface (API) calls, and dynamic link libraries (DLLs). It was also observed that N Gram technique can be used to detect important sequences in a ransomware attack. For example, a Registry Delete operation, whereby a malicious file attempts to delete registries, follows a particular and repeated sequence. A different observation involving benign files showed that Registry Delete operation does not follow any particular or repeated sequence. A reliable and efficient ransomware detection leverages on the nonexistence of a common Registry deleted sequence used by both malicious and benign files. Table 5 summarizes previous researches which proposed artificial neural network approaches for ransomware detection.

Table 5: Summary of related works (artificial neural network approaches)

Author	Problem addressed	Method used	Result
Agrawal (2019)	Establishing a relationship among events which follow a particular sequence	Attended Recent Inputs-Long Short-Term Memory (ARI-LSTM)	High detection accuracy for sequence learning models
Schmidhubar & Cummins, (1997; Gers (2000)	Establishing a relationship among events which follow a particular sequence	Attended Recent Inputs-Long Short-Term Memory (ARI-LSTM)	Have better detection rate than LSTM. False Positive Rate (FPR) set of 2%.
Arslan (2020)	Using unique features of ransomware to detect an attack	Transfer learning based deep convolutional neural networks	Detects some attributes, states, and patterns of ransomware files.

4.1.2 Non-Artificial Intelligence-Based Methods

Non-AI methods use approaches such as packet inspection and traffic analysis to detect ransomware. One of such methods aims at detecting ransomware using a network of decoy and bogus computer systems known as honeypot. The goal was to create and monitor honeypot folder for changes that could be used to detect the presence of ransomware (Moore, 2016). The study performed the manipulation of the Windows Security logs using the File Screening service of the *Microsoft File Server Resource Manager* feature and *EventSentry*. Although honeypot is a useful tool for tracking network activity, the method offers a limited view of ransomware and their activities on the network as the absence of attack alerts does not mean that a honeypot is not a target of ransomware attack. A related work proposed an algorithm that probes networks for passive monitoring of traffic in order to detect the presence of ransomware (Morato et al., 2018). Experimental analysis using 19 different ransomware families show that it takes the proposed algorithm less than 20 seconds to detect the presence of ransomware. It was also observed that not more than 10 files are lost within the 20 second duration. The method allows recovery of lost files as their contents were stored in the network traffic. It also has low false positives based on experiments conducted on traffic data from real-life corporate networks. A very recent neural network approach to ransomware detection is the novel Bayesian Neural network known as the Radial Spike and Slab Bayesian Neural Network (Nazarovs et al., 2022). The proposed solution is suitable for large and/or complex architectures as it provides better performance than the generic Bayesian Neural Network and other deep learning techniques. It also provides enough information to trigger the suspicion of investigators and confirm whether an incident is actually a ransomware attack or not. Overall, the technique helps to overcome the limitation of insufficient ransomware datasets for deep learning experiments by eliminating the likelihood of overfitting even if small-sized samples are used for training and classification. A limitation of the approach is the need for human intervention to disable systems and prevent network access in the event of a suspected ransomware attack. A summary of previous studies based on non-AI techniques is presented in Table 6.

Table 6: Summary of related works (non-artificial neural network approaches)

Author	Problem addressed	Method used	Result	Limitation
Moore (2016)	Ransomware Detection	Honeypot	N/A	Method offers a limited view of ransomware and their activities on the network
Morato et al. (2018)	Detecting the presence of ransomware and preventing attacks	N/A	Less than 20 seconds to detect the presence of ransomware.	N/A
Cabaj et al. (2017)	Software-Defined Network (SDN) environment	Rapid response to ransomware threats	Detection rates of between 97% and 98% as well as 4–5% false alarm rates	N/A
Chen et al. (2018)	Systematic characterization and real-time detection of Android ransomware	Novel technique for real time detection of encrypting ransomware	Abnormal encryption activities can be detected before a ransomware causes significant damages. The analysis of runtime performance also demonstrated the usability of ransomprober	Attempt at detecting mobile ransomware is constrained by the unavailability of a comprehensive dataset and limited understanding of real-time ransomware attack.
(Kharraz et al., 2015)	HELDROID	Distinguish known and unknown scareware and ransomware samples from goodware	Provides reliable protection against many zero-day ransomware attacks	N/A

A slightly different detection method used the modes of ransomware communication in a Software-Defined Network (SDN) environment to provide a rapid response to ransomware threats (Cabaj et al., 2017). The proposed method observes the network communication patterns of CryptoWall and Locky ransomware families between an infected host and an attacker's command and control server. Threat detection involves an analysis of the HTTP message sequences and the sizes of their respective contents. The results of experiments based on actual ransomware data showed high detection rates of between 97% and 98% as well as 4–5% false alarm rates. This shows that the approach is simple, realist and effective in preventing ransomware attacks. Chen et al., (2018) proposed a novel real-time detection system called RansomProber, which analyzes the user interface widgets of related activities and the coordinates of users' finger movements. The technique is suitable for a systematic characterization and real-time detection of Android ransomware. The results of the analysis of these samples from different perspectives revealed details such as the ransomware scale, classification, and features. The study also designed a novel technique for real time detection of encrypting ransomware. The goal is to monitor a device's sensitive files and determine the user's intention. The technique can accurately and reliably detect whether a file encryption activity initiated by users or ransomware. Experimental evaluation showed that proposed method can detect abnormal encryption activities before a ransomware causes significant damages. The analysis of runtime performance also demonstrated the usability of RansomProber. However, attempt at detecting mobile ransomware is constrained by the unavailability of a comprehensive dataset and limited understanding of real-time ransomware attack. A related work (Kharraz et al., 2015) proposed a mobile ransomware detection approach known as HELDROID to distinguish known and unknown scareware and ransomware samples from goodware in a quick, efficient and fully automated manner. The approach monitors abnormal file system behaviour to offer protection against a large number of ransomware. It also provides reliable protection against many zero-day ransomware attacks by examining I/O requests and protecting master file table in the NTFS file system. A very recent non-AI technique addressed the limitations of entropy-based ransomware detection such as misclassification due to high-level entropy of some legitimate files and impracticality of a wholesome evaluation of large files to detect

ransomware due to high cost of such effort (Kim et al., 2022). This was achieved by proposing EntropySA and DistSA as byte-frequency-based indicators which explore the properties of “sample areas" (SAs) of suspicious files. The discriminant feature used to distinguish a benign file from an infected file is the degree of randomness of information in the sampled sub-area of the files. An experimental evaluation of the proposed method showed that benign files whose sampled area includes information such as file header have relatively low degree of randomness despite the high level of randomness exhibited by the entire file. The main advantage of the approach is its ability to detect a ransomware based on each file it attacks. This makes the technique able to provide effective and accurate detection of ransomware attacks irrespective of the order in which a ransomware attacks files in the system. It is also robust against obfuscating ransomware which hide their behaviour to evade detection. However, the approach is unable to record 100% detection of files attacked by the DMA Locker2 ransomware because the ransomware places a unique signature string at the beginning of a file in order to evade detection. It is also unable to detect smaller (less than 256 bytes) files.

5 Prevention, Mitigation and Recovery Strategies

It is not only necessary to detect a ransomware attack after it has caused significant damages to data and systems, but also important to put strategies in place to prevent attacks from occurring. This makes it critical to devise approaches for preventing the occurrence of ransomware attacks and mitigating potential damages caused by ransomware. It is also important to ensure recovery of files and systems after attacks without any need to pay ransom. One of such methods focuses on preventing ransomware and protecting computer systems by identifying and blocking an attack (Patel & Tailor, 2020). The strategy involves fooling an attacker to encrypt a large dummy file over a long period of time. This provides sufficient time to render the remaining contents of the file system inaccessible to the ransomware. Performance evaluation of the proposed technique in a real-time environment showed that the approach is effective against ransomware attacks. A similar study used the behaviour of a system under advanced Petya ransomware attack to propose strategies for minimizing the susceptibility of systems and organizations to ransomware attacks (Aidan et al., 2018). The approach prevents Petya ransomware attack by blocking the server message block (SMB) ports (that is, UDP port 137, 138 and TCP 139, 445) or disabling SMBv1. Additional measure includes preventing the execution of perfc.dat and psexec.dat files from sysinternals. Perfc.dat and psexec.dat files are created as a result of ransomware attack. It is possible to prevent the creation of the ransomware files by self-creating perfc.dat and psexec.dat files and changing their access permissions to READONLY. Other mitigation strategies include using Software Restriction Policies (SRP) to disable binaries from executing %APPDATA%, %PROGRAMDATA% and %TEMP% paths, as well as restricting malicious files by deploying email and web filtering on the network. File- and behavior-based detection methods do not have the ability to detect or prevent previously unknown ransomware variants and ransomware which attack cloud-based data storage. This challenge was addressed by proposing a machine learning technique for ransomware prevention known as file entropy analysis (Lee et al., 2019). The method can retrieve infected files that have been synchronized to the backup server whether or not the host system is infected by ransomware. Similarly, Du et al. (2022) presented a number of defensive strategies which are able to detect a ransomware before it actually attacks an endpoint system. One of such is a hybrid machine learning solution based on intelligent KNN and density-based algorithms. The approach integrates data pre-processing and feature engineering techniques with KNN algorithm. It has high ransomware attack prediction accuracy of 98%, which makes it a suitable anti-malware and anti-ransomware solution. Another method used in the study is random forest which records a good accuracy of 99%. The study proposed K-means and DBSCAN clustering algorithms to provide effective detection of previously unknown ransomware variants. A very recent preventive solution is the system-architecture-based risk transference which relocates sensitive data from the system to highly protected storage locations (Sreejith Gopinath & Aspen Olmstead, 2022). This minimizes the susceptibility of such data to ransomware attacks. The information is also stored in a context-free manner in order to discourage attackers from attempting to hold such data hostage. Experimental results show that the proposed architecture allows for easy recovery of a system under ransomware attack.

The method proposed by Gómez-Hernández et al. (2022) supports early reaction to ransomware incidents and reduces damage to files during an attack. It is an enhanced tool which deploys a large number of “honey files” in close proximity to sensitive system and application files in order to achieve early detection of and timely response to ransomware attacks. The capability of the tool was extended by adapting it to Windows platforms and improving the system-wide management of the “honey files” to provide adequate protection of system files. Additional enhancements include semi-automation of defence mechanisms against ransomware using dynamic white-/black-lists, which minimizes the need for human intervention in the event of an attack. WmRmR (weighted minimum Redundancy maximum Relevance) is a mitigation strategy used to estimate the importance of dominant or most discriminating features in data captured at the onset of ransomware attacks (Ahmed et al., 2022). It is a hybrid solution based on the integration of two different techniques namely, enhanced minimum redundancy

maximum relevance (EmRmR) and Term Frequency-Inverse Document Frequency (TF-IDF). The approach uses TF-IDF to evaluate weights generated by EmRmR algorithm and eliminate noisy features that may impair performance. The results of experiments show that the proposed solution has simple implementation, low false positive rate and is effective for early detection of ransomware attacks.

A simple technique for easy recovery from ransomware attacks irrespective of the availability of attacker's tools on the victim system to prevent recovery from such attacks has also been proposed (Wecksten, 2016). The study revealed that common cypto ransomware attack involves the installation of tools on a victim's device to make recovery from ransomware attack a herculean task. Hence the need for a technique to provide easy recovery from ransomware infections by renaming the system tool which handles shadow copies of files. A similar strategy for ransomware recovery proposed by Kim et al. (2022) enables a partial (95%) recovery of the master key used by attackers to launch Hive ransomware. This was achieved by analyzing the encryption process used by Hive ransomware and discovering its vulnerabilities. The result of this effort is the creation of a decryption key for recovering files held by the ransomware without the need to obtain the attacker's RSA private key or pay a ransom to the attacker. A very recent recovery method is the novel framework proposed as an efficient technique for recovering XML documents that have been compromised by ransomware attack (Al-Dwairi et al., 2022). The approach uses the concepts of links to support the distributed storage of different versions of the same file. Adequate access control is also put in place to prevent the file versions from unauthorized encryption or deletion. Experimental results show that the time required for decrypting an encrypted XML file is directly proportional to the actual size of the file before encryption. Generally, files that less than 1 MB requires less than 120 ms and decryption of bzip2 encrypted files required the highest CPU utilization. Decrypting zip and gzip encrypted files requires almost the same amount of memory (~ 6.8 KBs), while decryption of bzip2 encrypted files increases the memory usage to 28 KBs. Overall, the approach is efficient in terms of storage overhead, processing time, CPU utilization, and memory usage.

6 Future Research Directions

Path enumeration for creation of decoy file proposed by Lalson et al. (2019) takes several hours in very large file systems. Hence, there is a need to maintain a balance between the file size and the computation time for creating large decoy files. The threshold can be tweaked to suit the peculiarities of each system. For example, a high threshold may be used in critical systems to minimize the false positive rate, while home systems may have threshold values lower than those of critical systems. Future research works should also consider enhancing the technique for detecting multi-stage crypto ransomware attacks suggested by Zimba et al. (2018) to prioritize the security of production network devices using a cascaded network segmentation approach. Research effort should also concentrate on detecting network-level ransomware attacks because many ransomware now communicate with the command-and-control server via encrypted channels such as the HTTPS protocol. The work of Makinde et al. (2019) is limited by the fact that the simulation involved few users. A future work should focus on using tools for big data analytics to simulate the behaviour of large number of users. The solution proposed by Sheen and Yadav (2018) applied class imbalance due to unequal number of samples in ransomware dataset and benign dataset. The same technique should be applied on a balanced dataset using the same classifiers and observe the outcome.

Although the Deep Packet Inspection technique proposed in Aragom et al. (2016) has 93% accuracy, the model currently supports static analysis. It can be extended to handle dynamic analysis by implementing it on a software defined network to support real time ransomware detection. Another possible extension is to improve the feature selection process applied to pcap files, such that the enhanced method compares the extracted features with those obtained from preceding or successive packets in order to obtain a better detection rate. The results obtained from the simple MLP network proposed in Vinayakumar et al. (2017) do not represent the actual situation involving more complex architecture settings. Future works should focus on using more complex MLP networks to perform the same experiments on state-of-the-art hardware in a distributed environment. Zahra and Sha (2017) proposed an IoT ransomware detection technique without actual implementation and deployment in a real-world environment. The proposed technique should be prototyped and deployed in a real-world IoT environment in order to evaluate and refine it. A future work should focus on increasing the accuracy of Randroid proposed by Alzahrani et al. (2018) by adding of more samples of malicious images and strings to the ISM database and the SSM database respectively. The new images should include logos of governments and icons of law enforcement agencies. This will help detect ransomware variants that exploit false positives such as fake FBI notes. Additional consideration should be given to the application of text recognition techniques on more images and texts to verify the ability of the dynamic analysis component to detect dynamic payloads. Chen et al. (2018) suggested that detecting mobile ransomware is constrained by the unavailability of a comprehensive dataset and limited understanding of real-time ransomware attack. Future research should consider creating a comprehensive and up-

to-date dataset of mobile ransomware and developing a deep understanding of real-time ransomware attack against mobile devices.

Recent studies also have limitations and gaps which may be explored by future research. Future research based on the work of Rani and Dhavale (2022) should consider the integration of the model with Elasticsearch Logstash Kibana (ELK) to develop a practical tool for real-life ransomware detection. ELK can serve as the backend for filtering and collecting useful log data for the ransomware detection system. The detection system will then process logs of suspicious activities to determine whether such events are actually ransomware attacks. The work of Ahmed et al. (2022) focused only on the use of static and dynamic features for detecting unknown attacks by malicious Android malware. The study can be extended to explore distinct and detailed features of known ransomware samples, attacks that can be launched by such ransomware samples, qualitative and economical strategies for feature extraction, and malicious feature estimation. Researchers may also propose suitable metrics to determine the resistance of ransomware against countermeasures as well as the performance of defence mechanisms against ransomware attacks. The inability of the byte-frequency-based indicators proposed by Kim et al. (2022) to detect smaller (less than 256 bytes) files also represents an important research problem. This is because attackers may evade detection by using small-sized ransomware files to exploit computer systems. The approach can also be enhanced to address its inability to detect the DMA Locker2 ransomware. The novel Bayesian Neural network known as the Radial Spike and Slab Bayesian Neural Network (Nazarovs et al., 2022) requires human intervention for disabling and isolating systems in the event of ransomware attack. Future works should explore an enhanced solution which automatically disables systems and prevent access to the network once there is a suspected ransomware attack. Techniques which use enhanced feature extraction methods for better ransomware detection also require improvements. The two-stage particle swarm optimization proposed by Abbasi et al. (2022) requires improvements such as the use of more feature sets in the experimental dataset to capture additional behavioral traits such as communication involving critical servers or command and control centre. Also, certain future sets may be removed from the dataset and observe the impact of such removal on performance. In addition to this, intending researchers may explore the use system call sequences as additional features for classifying ransomware into families.

7 Conclusion

Ransomware attacks have done and are still doing significant damages to computers as well as data and information they process. These include unauthorized access, disclosure and destruction of vital, sensitive and critical computer and hardware resources. Both individuals and corporations have suffered grave financial losses and reputational damages due to ransomware attacks. Hence, several methods have been proposed for accurate, timely and reliable ransomware detection techniques. The background discussion on ransomware as well as the historical background and chronology of ransomware attacks presented in this study provide readers with the much-needed introduction to ransomware detection. The detailed and critical review of recent papers provide readers with an up-to-date knowledge of the current trends in automated ransomware detection. This will equip readers with the knowledge of the state-of-the-art in automated ransomware detection, prevention, mitigation and recovery. Also included in this study is an exposé on future research directions to provide intending researchers with open issues and possible research problems in detection, prevention, mitigation of and recovery from ransomware attacks.

References

Acronis International (2021). How machine learning can be used to prevent ransomware. Retrieved from https://www.acronis.com/en-eu/articles/machine-learning-prevent-ransomware.

Adamov, A. & Carlsson A. (2017). The state of ransomware. Trends and mitigation techniques. IEEE East-West Design & Test Symposium (EWDTS), 1-8, doi: 10.1109/EWDTS.2017.8110056.

Adamu, U. & Awan, I. (2019). Ransomware prediction using supervised learning algorithms. FiCloud 2019, Istanbul, Turkey, 57–63. doi: 10.1109/FiCloud.2019.00016.

Agrawal R., Stokes J.W., Selvaraj K. & Marinescu, M. (2019). Attention in recurrent neural networks for ransomware detection. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3222-3226, doi: 10.1109/ICASSP.2019.8682899.

Ahmad, A., Kaiiali, M., Sezer, S. & O'kane P. (2019). A multi-classifier network-based crypto ransomware detection system: a case study of locky ransomware. IEEE Access, vol. 7, doi: 10.1109/ACCESS.2019.2907485.

Ahmed, U., Lin J.C.W. & Srivastava, G. (2022). Mitigating adversarial evasion attacks of ransomware using

ensemble learning. Computers and Electrical Engineering, 100 (2022) 107903.

Ahmed Y.A., Huda S., Al-rimy B.A.S., Alharbi N., Saeed F, Ghaleb F.A. & Ali I.M. (2022). A weighted minimum redundancy maximum relevance technique for ransomware early detection in industrial iot sustainability. MDPI. 14(1231), 1-15. Retrieved from https://doi.org/10.3390/su14031231.

Aidan J., Zeenia, S. & Garg, U. (2018). Advanced petya ransomware and mitigation strategies. First International Conference on Secure Cyber Computing and Communication (ICSCCC). 23-28, doi: 10.1109/ICSCCC.2018.8703323.

Al-Dwairi M., Shatnawi A.S., Al-Khaleel, O. & Al-Duwairi, B. (2022). Ransomware-resilient self-healing XML documents. Future Internet, 14(115), 1-19. Retrieved from https://doi.org/10.3390/fi14040115.

Alzahrani A. (2018). RanDroid: structural similarity approach for detecting ransomware applications in android platform. IEEE International Conference on Electro/Information Technology (EIT), 0892-0897. doi: 10.1109/EIT.2018.8500161.

Ameer, M. (2019). Android Ransomware Detection using Machine Learning Techniques to Mitigate Adversarial Evasion Attacks. (Capital University of Science and Technology, Islamabad, Pakistan).

Andronio N., Zanero S. & Maggi F. (2015). HelDroid: dissecting and detecting mobile ransomware. In Research in Attacks, Intrusions, and Defenses. Lect. Notes Comput. Sci., vol. 9404, 382–404.

Aragorn, T., Yun-chun, C., YiHsiang, K., & Tsungnan, L. (2016). Deep learning for ransomware detection. Retrieved from https://www.semanticscholar.org/paper/Deep-Learning-for-Ransomware-Detection-Aragorn-Yun-chun/cc3a41b37230861cfe429632744e0d1db19256b7.

Arslan A., Abdul A., Umme Z., & Asifullah, K. (2020). Ransomware analysis using feature engineering and deep neural networks. Retrieved from https://arxiv.org/abs/1910.00286v2.

Azmoodeh A., Dehghantanha A., Conti M, & Choo K. R (2018). Detecting crypto Ransomware in IoT networks based on energy consumption footprint. Ambient Intell Human Comput 9, 1141–1152, Retrieved from https://doi.org/10.1007/s12652-017-0558-5.

Bazrafshan, Z., Hashemi, H, Fard, S.M.H. & Hamzeh, A. (2013). A survey on heuristic malware detection techniques. The 5th Conference on Information and Knowledge Technology, 113-120, doi: 10.1109/IKT.2013.6620049.

Brewer, R. (2016), Ransomware attacks: detection, prevention and cure. Netw. Secur, 1–6.

Cabaj, K., Gregorczyk, M., & Mazurczyk, W. (2017). Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. Comput. Electr. Eng., 353-368.

Celdrán A.H, Sánchez P.M.S, Castillo M.A, Gérôme B, Gregorio M.P. & Burkhard S (2022). Intelligent and behavioral-based detection of malware in IoT spectrum sensors. Int. J. Inf. Secur, 1-21. Retrieved from https://doi.org/10.1007/s10207-022-00602-w.

Chen, J., Wang, C., Zhao, Z., Chen, K., Du, R. & G.-J. Ahn (2018). Uncovering the face of android ransomware: characterization and real-time detection. IEEE Trans. Inf. Forensics Secur. 1286–1300.

Crowdstrike (2022a). How ransomware works. Retrieved from https://www.crowdstrike.com/resources/infographics/

how-fileless-ransomware-works/

Crowdstrike (2022b). Fileless Malware Explained. Retrieved from https://www.crowdstrike.com/cybersecurity-101/malware/fileless-malware/

Dargahi, T., Dehghantanha, A., Bahrami, P. N., Conti, M., Bianchi, G., & Benedetto, L. (2019). A cyber-kill-chain based taxonomy of crypto-ransomware features. Journal of Computer Virology and Hacking Techniques, 15(4), 277-305. Retrieved from https://doi.org/10.1007/s11416-019-00338-7.

Dehghantanha, A., Baldwin, J., & Alhawi. O. M. K. (2018). Leveraging machine learning techniques for windows ransomware network traffic detection. Retrieved from https://doi.org/10.1007/978-3-319-73951-95.

Dontov, D. (2019). Ransomware detection using machine learning. Retrieved from https://spinbackup.com/blog/ransomware-detection-using-machine-learning/

Du, J., Raza, S.H., Ahmad, M., Alam, I., Dar, S.H, & Habib, M.A, (2022). Digital forensics as advanced ransomware pre-attack detection algorithm for endpoint data protection. Security and Communication Networks. 1-16. Retrieved from https://doi.org/10.1155/2022/1424638.

eScan (2017). Antivirus reports.

F-Secure Labs (2013). Threat Report H1, Helsinki, Finland.

Fingers, J. (2020). Ransomware may have led to the death of a German hospital patient. Retrieved from www.google.com/amp/s/www.engadget.com/amp/ransomware-death-at-german-hospital-210309749.html.

Fitzpatrick, D. & Griffin, D. (2016). Cyber-extortion losses skyrocket, says FBI. Retrieved from http://money.cnn.com/2016/04/15/technology/ransomwarecyber-security.

Gallegos-Segovia, P.L., Bravo-Torres, J.F., Larios-Rosillo, V.M., Vintimilla-Tapia, P.E., Yuquilima-Albarado, I.F.

& Jara-Saltos J.D. (2017). Social engineering as an attack vector for ransomware. CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 1-6, doi: 10.1109/CHILECON.2017.8229528.

Gers, F.A., Schmidhuber, J. & Cummins, F.A (2000). Learning to forget: Continual prediction with lstm, Neural Computation. Neural Comput 2000. 12(10) 2451–2471. Retrieved from https://doi.org/10.1162/

089976600300015015

Gómez-Hernández, J.A., Sánchez-Fernández, R. & García-Teodoro, A. (2022). Inhibiting crypto-ransomware on windows platforms through a honeyfile-based approach with R-Locker. IET Inf. Secur. 16(1), 64–74. Retrieved from https://doi.org/10.1049/ise2.12042.

Gopinath, S. & Olmstead, A. (2022). Mitigating the effects of ransomware attacks on healthcare systems.

Hwang J, Kim J, L. S, & Kim K (2020). Two-stage ransomware detection using dynamic analysis and machine learning techniques. Wireless Pers Commun 112, 2597–2609, Retrieved from https://doi.org/10.1007/s11277-020-07166-9.

Jasmin, M. (2019). Detecting ransomware in encrypted network traffic using machine learning. (University of Victoria, Canada). Retrieved from http://hdl.handle.net/1828/11076.

Juan, A., Silver, H., & Hernández-Alvarez, M. (2017). Ransomware detection by cognitive security, IEEE, 346–363.

Khammas, B. (2020). Ransomware detection using random forest technique. ICT Express, 6(4), 325–331.

Khammas, B.M. (2022). Comparative analysis of various machine learning algorithms for ransomware detection. TELKOMNIKA Telecommunication Computing Electronics and Control, 20(1), 43~51.

Kharraz A., Robertson W, Balzarotti D, Leyla Bilge & Kirda E (2015). Cutting the gordian knot: a look under the hood of ransomware attacks In: M. Almgren., V. Gulisano, F. Maggi. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA Lecture Notes in Computer Science, vol 9148. Springer, Cham. Retrieved from https://doi.org/10.1007/978-3-319-20550-2_1.

Kim, G., Kim, S., Kang, J. & Kim, J. (2022). A method for decrypting data infected with hive ransomware. arXiv:2202.08477v1 [cs.CR], 1-23.

Kim, G.Y., Paik J.Y. & Kim Y. (2022). Byte frequency-based indicators for crypto-ransomware detection from empirical analysis. Journal of Computer Science and Technology, 37(2). DOI 10.1007/s11390-021-0263-x.

Lalson, E.R., Shony, K.M, & Netto, D.F. (2019). An integrated approach for detecting ransomware using static and dynamic analysis. FiCloud 2019, 410–414. doi: 10.1109/FiCloud.2019.00016.

Lee, K., Lee, S,, & Yim, K, (2019). Machine learning based file entropy analysis for ransomware detection in backup systems. IEEE Access, 110205–110215, doi: 10.1109/ACCESS.2019.2931136.

Lee, S., Jho, N., Chung D, Kang, Y. & Kim, M. (2022). Rcryptect: real-time detection of cryptographic function in the user-space filesystem. Computers & Security. 112, 1-13.

Makinde, O., Sangodoyin, A., Mohammed, B., Neagu, D., & Adamu, U. (2019). Distributed network behaviour prediction using machine learning and agent-based micro simulation. FiCloud 2019, 182-188.

Maniath S, Ashok A., Poornachandran P., Sujadevi G., Sankar,. A.U. & Jan, S (2017). Deep learning LSTM based ransomware detection. Recent Dev. Control Autom. Power Eng., 442–446, doi: 10.1109/RDCAPE.2017.8358312.

Matthias, H. (2018). Detecting ransomware. (Universität Konstanz).

McIntosh, T., Kayes, A.S.M., Chen, Y.P.P., Ng, A. & Watters, P, (2021). Ransomware mitigation in the modern era: a comprehensive review, research challenges, and future directions. ACM Computing Surveys (CSUR), 54(9), 1-36. Retrieved from https://doi.org/10.1145/3479393.

Microsoft Ignite (2022). What is ransomware? Retrieved from https://docs.microsoft.com/en-us/security/compass/human-operated-ransomware.

Mohurle, S., & Patil, S. (2017). Brief study of wannacry ransomware attack. Int. J. Adv. Res. Comput. Sci., vol. 8, 1938–1940.

Moore, C. (2016), Detecting ransomware with honeypot techniques. Cybersecurity and Cyberforensics Conference (CCC). 77-81. doi: 10.1109/CCC.2016.14.

Morato, D., Berrueta, E., Magaña E., & Izal, M. (2018). Ransomware early detection by the analysis of file sharing traffic. J. Netw. Comput. Appl., 14–32.

Nazarovs, J., Stokes, J.W, Turcotte, M., Carroll, J. & Grady, I. (2022). Radial spike and slab bayesian neural networks for sparse data in ransomware attacks. arXiv:2205.14759v1 [cs.CR] 1-17.

Olani, G., Wu, C-F. & Chang, Y-H. (2022). DeepWare: imaging performance counters with deep learning to detect ransomware. IEEE Transactions on Computers, Vol. X, No. X, XXX 20XX, pp. 1-15.

Oz, H., Aris, A., Levi, A., & Uluagac, A. S. (2021). A survey on ransomware: evolution, taxonomy, and defense solutions. ACM Computing Surveys (CSUR). Retrieved from https://doi.org/10.1145/3514229.

Patel, A. & Tailor, J, (2020). A malicious activity monitoring mechanism to detect and prevent ransomware. Comput. Fraud Secur, 14–19.

Potoroaca, A. (2020). Over 41% of cyber insurance claims in 2020 came from ransomware attacks. Retrieved from https://www.techspot.com/amp/news/86714-over-41-percent-cyber-insurance-claims-2020-came.html.

Poudyal, S., Subedi, K.P. & Dasgupta, D. (2018). A framework for analyzing ransomware using machine learning. IEEE Symposium Series on Computational Intelligence (SSCI), 1692-1699. doi: 10.1109/SSCI.2018.8628743.

Rahman, M. & Hasan, M. (2017). A support vector machine-based ransomware analysis framework with integrated feature set. 20th International Conference of Computer and Information Technology, Dhaka, 1–7. doi: 10.1109/ICCITECHN.2017.8281835.

Rani, N. & Dhavale, S.V. (2022). Leveraging machine learning for ransomware detection. arXiv:2206.01919v1 [cs.CR], 1-13.

Ransomware attacks. (2021). Top 5 ransomware attacks to watch out for in 2020-2021. Retrieved from https://www.google.com/amp/s/top-5-ransomware-attacks-to- watch-out-for-in-2020-2021/amp.

Richardson, R. & North, M. (2017). Ransomware: evolution, mitigation and prevention. Int. Manag. Rev., vol. 13, 10–21.

Savage, K., Coogan P, & Lau, H. (2015). The evolution of ransomware. Secur. Response, Symantec. Retrieved from https://its.fsu.edu/sites/g/files/imported/storage/images/information-security-and-privacy-office/the-evolution-of-ransomware.pdf.

Scaife, N., Carter, H., Traynor, P, & Kevin, B. (2016). CryptoLock (and drop it): stopping ransomware attacks on user data. IEEE 36th Int. Conf. Distrib. Comput. Syst.

Schmidhuber, J. & Sepp, H. (1997). Long short term memory. Neural Computation. 1735–1780.

Sgandurra D., Muñoz-González, L., Mohsen, R., & Lupu, E. (2016). Automated dynamic analysis of ransomware: benefits, limitations and use for detection. Retrieved from https://arxiv.org/abs/1609.03020, 1–12.

Sharmeen, S., Ahmed, Y.A., Huda, S., Koçer, B.S., & Hassan, M.M. (2020). Avoiding future digital extortion through

robust protection against ransomware threats using deep learning based adaptive approaches. IEEE Access, vol. 8, 24522–24534, doi: 10.1109/ACCESS.2020.2970466.

Shaukat, S., & Ribeiro, V. (2018). RansomWall: a layered defense system against cryptographic ransomware attacks using machine learning. 10th International Conference on Communication Systems and Networks, 356-363.

Sheen, S. & Yadav, A. (2018). Ransomware detection by mining api call usage. International Conference on Advances in Computing, Communications and Informatics (ICACCI), 983-987, doi: 10.1109/ICACCI.2018.8554938.

Singh, A., Ikuesan, R.A. & Venter, H. (2022). Ransomware detection using process memory. ICCWS 2022: 17th International Conference on Cyber Warfare and Security, 1-10.

Symantec Corporation (2016). Internet security threat report.

Talabani, H.S. & Abdulhadi, H.M.T. (2022). Bitcoin ransomware detection employing rule-based algorithms.

Science Journal of University of Zakho, 10(1), 5– 10.

Vehabovic, A., Ghani, N., Bou-Harb, E., Crichigno, J. & Yayimli, A. (2022). Ransomware detection and classification strategies. IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 316-324, doi: 10.1109/BlackSeaCom54372.2022.9858296.

Vinayakumar, R., Soman, K.P., Senthil, M., Velan, K. K. & Ganorkar, S. (2017). Evaluating shallow and deep networks for ransomware detection and classification. International Conference on Advances in Computing, Communications and Informatics (ICACCI), 259-265. doi: 10.1109/ICACCI.2017.8125850.

Wan, Y., Chang, J., Chen, R. & Wang, S. (2018). Feature-selection-based ransomware detection with machine learning of data analysis. 3rd International Conference on Computer and Communication Systems (ICCCS), 85-88, doi: 10.1109/CCOMS.2018.8463300.

Weckstén, M., Frick, J., Sjöström, A. & Järpe, E. (2016). A novel method for recovery from crypto ransomware infections. 2nd IEEE International Conference on Computer and Communications (ICCC). 1354-1358, doi: 10.1109/CompComm.2016.7924925.

Wongsupa, P. (2018). Deep learning for android application ransomware detection. MSc Dissertation. (Florida Atlantic University).

Yang, T., Yang, Y., Qian K., Lo, D.C, Qian, Y. & Tao, L. (2015). Automated detection and analysis for android

ransomware. IEEE 17th International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security, and IEEE 12th International Conference on Embedded Software and Systems, 1338-1343, doi: 10.1109/HPCC-CSS-ICESS.2015.39.

Zahra, A. & Shah, M. (2017). IoT based ransomware growth rate evaluation and detection using command and control blacklisting. Proceedings of the 23rd International Conference on Automation & Computing, (University of Huddersfield, Huddersfield), 1–6.

Zetter, K. (2015). Hacker lexicon: A guide to ransomware, the scary hack that's on the rise. Retrieved from: https://www.wired.com/2015/09/hacker-lexicon-guideransomware- scary-hack-thats-rise/

Zimba, A., Wang, Z., & Chen, H. (2018). Multi-stage crypto ransomware attacks: a new emerging cyber threat to critical infrastructure and industrial control systems. ICT Express, vol. 4, 14–18.