

---

People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research  
University of Saad Dahleb – Blida 1  
Faculty of Sciences  
Department of Computer Science

---

# Master Thesis



In Computer Science

Option: Software Engineering



## Ransomware detection using Deep Learning

Authors:

**BELHADJ Akram Djalal & HAMID SIDI YKRELEF Abdelfettah**

**Mrs. YKHLEF Hadjer**

**President**

**Mrs. BERRAMDANE Djamila**

**Examinator**

**Mr. CHIKHI Nacim Fateh**

**Supervisor**

**June 2023**

## **Acknowledgements**

In the name of Allah, the most gracious and the most merciful. First and foremost, I am thankful to the Almighty Allah for giving me the strength, knowledge, ability and opportunity to undertake this work.

Writing this thesis definitely took a toll on me mentally but remembering that surely with hardship comes ease helped me keep a sane mind and commit to my work.

We would like to express our deepest and most sincere gratitude to our supervisor Mr. CHIKHI for giving us the opportunity to conduct this research and providing us invaluable guidance throughout our work.

Last but not least, we would like to express our heartfelt gratitude to our family for their unwavering support and encouragement throughout our university careers and lives.

Their love and understanding have been invaluable in guiding us through this breathtaking journey. We would also like to extend our gratitude to our friends and colleagues who joined us in this extraordinary experience. Their company and the spirit of camaraderie they brought to our trip will live with us forever.

## *Abstract*

Ransomware is malicious software that encrypts victims' data and demands a ransom to decrypt them. This type of malware attacks are becoming more sophisticated, posing a significant threat to individuals and organizations. This research focuses on developing a powerful ransomware detection model that integrates behavioral analysis, deep learning, and bootstrapping techniques. The model uses behavioral analysis to identify ransomware samples, while deep learning techniques train multiple specialized models to detect zero-day ransomware attacks and minimize false positives. The proposed model outperforms machine learning algorithms in terms of accuracy, precision, and recall. This work should serve as the first step for further research and exploration of additional features, behavioral indicators, static analysis techniques, and hybrid approaches to enhance detection capabilities and combat ransomware threats, and finally to deployment in production.

**Keywords:** Ransomware Detection, Deep Learning, Feedforward Neural Network, Machine Learning, Ensemble Learning.

## *Résumé*

Le rançongiciel est un logiciel malveillant qui crypte les données des victimes et exige une rançon pour les décrypter. Ce type d'attaques par malware devient de plus en plus sophistiqué, représentant une menace significative pour les individus et les organisations. Ce travail se concentre sur le développement d'un modèle de détection de rançongiciel puissant qui intègre l'analyse comportementale, l'apprentissage profond et les techniques de bootstrapping. Le modèle utilise l'analyse comportementale pour identifier les échantillons de rançongiciel, tandis que les techniques d'apprentissage profond forment plusieurs modèles spécialisés pour détecter les attaques de nouveaux ransomware et minimiser les faux positifs. Le modèle surpasse les algorithmes d'apprentissage automatique en termes de accuracy, de précision et de rappel. Ce travail devrait servir de première étape pour d'autres recherches et l'exploration de fonctionnalités supplémentaires, d'indicateurs comportementaux, de techniques d'analyse statique, et d'approches hybrides pour améliorer les capacités de détection et lutter contre les menaces de ransomware, et enfin pour le déploiement en production.

**Mots-clés :** Détection de Ransomware, Apprentissage Profond, Réseau de neurones à propagation avant, Apprentissage Automatique, Apprentissage ensembliste.

## ملخص

برامج الفدية هي برامج ضارة تقوم بتشفير بيانات الضحايا وتطلب فدية لفك تشفيرها. هذا النوع من هجمات أصبح أكثر تعقيدًا، مما يشكل تهديدًا كبيرًا للأفراد والمنظمات. تركز هذه الأبحاث على تطوير نموذج قوي لاكتشاف الفدية يدمج التحليل السلوكي، التعلم العميق، وتقنيات البوتسترايب. يستخدم النموذج التحليل السلوكي لتحديد عينات الفدية، بينما تدرّب تقنيات التعلم العميق عدة نماذج متخصصة لاكتشاف هجمات الفدية الغير مكتشفة وتقليل موجب خاطئ. يتفوق النموذج على خوارزميات التعلم الآلي في الدقة، والجودة، والاستدعاء. يجب أن يكون هذا العمل الخطوة الأولى لمزيد من البحث واستكشاف الميزات الإضافية، المؤشرات السلوكية، تقنيات التحليل الثابت، والنهج المختلطة لتحسين قدرات الكشف ومكافحة تهديدات الفدية، وأخيرًا للنشر في الإنتاج.

**الكلمات المفتاحية:** الكشف عن برامج الفدية، التعلم العميق، الشبكة العصبية أمامية التغذية، التعلم الآلي، التعلم الجماعي.

# *Acronyms*

<b>AES</b>	Advanced Encryption Standard
<b>API</b>	Application Programming Interface
<b>C2C</b>	Command & Control (C&C) Server
<b>CNN</b>	Convolutional Neural Network
<b>CPU</b>	Central Processing Unit
<b>DL</b>	Deep Learning
<b>FNN</b>	Feedforward Neural Network
<b>KNN</b>	K nearest Neighbor
<b>LR</b>	Logistic Regression
<b>LSTM</b>	Long Short-Term Memory
<b>ML</b>	Machine Learning
<b>NB</b>	Naïve Bayes
<b>OS</b>	Operating System
<b>PDF</b>	Portable Document Format
<b>RAAS</b>	Ransomware As A Service
<b>RDP</b>	Remote Desktop Protocol
<b>RNN</b>	Recurrent Neural Network
<b>RSA</b>	Rivest, Shamir et Adelman
<b>SGD</b>	Stochastic Gradient Descent
<b>SVM</b>	Support Vector Machine
<b>USB</b>	Universal Serial Bus
<b>VPN</b>	Virtual Private Network

# Table of contents

<b>Abstract .....</b>	<b>i</b>
<b>Résumé .....</b>	<b>i</b>
<b>Acronyms .....</b>	<b>iii</b>
<b>List of Figures .....</b>	<b>iv</b>
<b>List of Tables .....</b>	<b>vi</b>
<b>Introduction .....</b>	<b>1</b>
<b>Chapter 1 State of the art .....</b>	<b>2</b>
.1 Introduction to ransomware .....	2
.1.1 Ransomware definition.....	3
.1.2 Difference between ransomware and malware .....	3
.1.2.1 Malware.....	3
.1.2.2 Differences between malware and ransomware .....	3
.1.3 Evolution of ransomware.....	4
.1.3.1 The emergence of ransomware: A 1989-2006 Analysis.....	4
.1.3.2 The Changing Face of Ransomware: A 2010-2019 Analysis.....	4
.1.3.3 Recent ransomware (2020-2023).....	7
.1.4 Anatomy of ransomware attacks.....	8
.1.4.1 Ransomware attack vectors .....	8
.1.4.2 The Six Stages of Ransomware Attacks .....	10
.1.4.3 How Ransomware keeps your digital life as hostage .....	11
.1.4.4 Ransomware Component .....	12

.2	Types of ransomwares .....	14
.2.1	Locker .....	14
.2.1.1	How Locker behaves inside a system: .....	15
.2.2	Encryptor .....	15
.2.2.1	How Encryptor behaves inside a system: .....	15
.2.3	Leakware .....	15
.2.4	Other types of ransomware: .....	16
.2.4.1	Double extortion .....	16
.2.4.2	Scareware .....	17
.2.4.3	RaaS (ransomware as a service) .....	17
.3	Fundamental techniques for analyzing ransomware .....	17
.3.1	Static Analysis: .....	17
.3.2	Dynamic Analysis .....	18
.3.3	Hybrid analysis .....	19
.4	Traditional ransomware detection .....	20
.4.1	Signature-based detection methods: .....	20
.4.2	Behavior-based detection methods .....	21
.4.3	Heuristic-based methods .....	21
.5	Tradition solutions .....	22
.6	Conclusion .....	23
	<b>Chapter 2 Machine Learning and Deep Learning .....</b>	<b>24</b>
.1	Introduction .....	24
.2	Machine Learning .....	24
.2.1	Overview of Machine learning application in cybersecurity .....	24
.3	Famous ML algorithms used for ransomware detection: .....	24
.3.1	Support Vector Machines (SVM) .....	24

.3.2	Decision trees.....	25
.3.3	Random Forest.....	26
.4	Deep learning .....	27
.4.1	Types of neural networks.....	27
.4.1.1	FNN .....	27
.4.1.2	CNN .....	28
.4.1.3	RNN.....	29
.4.1.4	LSTM .....	30
.4.1.5	AutoEncoders .....	31
.5	Previous studies on ransomware detection.....	31
.6	Conclusion .....	34

**Chapter 3 Proposed approach ..... 35**

.1	Dataset Collection and Preprocessing.....	35
.1.1	Description of the dataset .....	35
.1.2	Preprocessing steps and feature extraction techniques applied to the data: .....	36
.1.3	Feature selection: .....	37
.1.3.1	ANOVA F_Value: .....	37
.1.3.2	Chi_Squared_Test:.....	37
.1.3.3	Mutual Information: .....	37
.2	Model Design.....	38
.2.1	Ensemble learning:.....	38
.2.1.1	Bagging (Bootstrap Aggregating):.....	38
.2.1.2	Boosting: .....	38
.2.1.3	Stacking:.....	38

.2.2	Proposed Method: .....	39
.3	Model Training and Validation .....	40
.4	Results: .....	42
.5	Implementation.....	44
.6	CONCLUSION: .....	49
	<b>Conclusion.....</b>	<b>50</b>

## *List of Figures*

<b>Figure 1.1</b> : Number of ransomware covered per survey article .	2
<b>Figure 1.2</b> : Ransomware damage over the recent year	2
<b>Figure 1.3</b> : AIDS trojan ransomware note	4
<b>Figure 1.4</b> : Reveton ransom note	5
<b>Figure 1.5</b> : WannaCry execution flowchart .	6
<b>Figure 1.6</b> : Evolution of major ransomware families from 2010 to 2019	7
<b>Figure 1.7</b> : New ransomware strains from 2020 to Q1 of 2023	8
<b>Figure 1.8</b> : Ransomware Attack vectors between 2018 and 2022 .	9
<b>Figure 1.9</b> : Djvu Ransomware’s sequence of operations .	13
<b>Figure 1.10</b> : Taxonomy of ransomware	14
<b>Figure 1.11</b> : Locker Ransom note	14
<b>Figure 1.12</b> : How Double extortion ransomware works	16
<b>Figure 1.13</b> : Potential traits (ransomware dataset features)	19
<b>Figure 1.14</b> : Taxonomy of Ransomware detection techniques .	19
<b>Figure 1.15</b> : Type of analysis distribution in Literature Studies	20
<b>Figure 2.1</b> : Classification of two tapes using SVM	25
<b>Figure 2.2</b> : Decision Trees	26
<b>Figure 2.3</b> : Generalized structure for random forest	27
<b>Figure 2.4</b> : Multi-layer perceptron (MLP-NN) basic Architecture	28
<b>Figure 2.5</b> : Basic architecture of CNN	29
<b>Figure 2.6</b> : Different representations of recurrent layer	30
<b>Figure 3.1</b> : Number of samples for each family of Ransomware	35
<b>Figure 3.2</b> : Different types of ensembles learning methods	38
<b>Figure 3.3</b> : Proposed method	39
<b>Figure 3.4</b> : Flowchart of base learners training process	41
<b>Figure 3.5</b> : User File Prediction Workflow: From Upload to Final Decision	44
<b>Figure 3.6</b> : Prediction mechanism using the report	45
<b>Figure 3.7</b> : Global Sequence diagram	45
<b>Figure 3.8</b> : Check Status Sequence Diagram	46
<b>Figure 3.9</b> : Windows sandbox _pic from askleo.com	43

<b>Figure 3.10</b> : Cuckoo sandbox logo.....	44
<b>Figure 3.11</b> :Upload a File for analysis .....	48
<b>Figure 3.12</b> : Software execution in the VM .....	48
<b>Figure 3.13</b> : Successful Classification: Ransomware (Tslacypt) and Goodware (7zip) .....	49

## *List of Tables*

<b>Table 1.1</b> : Differences between malware and ransomware .	3
<b>Table 1.2</b> : Vitality of ransomware .	11
<b>Table 1.3</b> : How locker behave inside a system.....	15
<b>Table 1.4</b> : How Encryptor behave inside a system.....	15
<b>Table 1.5</b> : How Leakware behave inside a system.....	16
<b>Table 2.1</b> : Previous studies comparison .....	33
<b>Table 3.1</b> : Dataset features classification. ....	36
<b>Table 3.2</b> : Train test split. ....	36
<b>Table 3.3</b> : Number of samples of training and testing sets.....	37
<b>Table 3.4</b> : Selected features per class. ....	37
<b>Table 3.5</b> : Training result of base learners and the final prediction model. ....	42
<b>Table 3.6</b> : Comparison with various ML algorithms and current techniques in the field. ....	42
<b>Table 3.7</b> : Cross validation results compared with various ML algorithms.....	43
<b>Table 3.8</b> : Analysis of Ransomware Samples and Detection Status. ....	47
<b>Table 3.9</b> : Analysis of Software Samples and Detection Results. ....	47

# *Introduction*

Ransomware has emerged as a significant cybersecurity concern, as it encrypts critical data or restricts computer access, often accompanied by blackmail demands for a ransom. Traditional detection approaches have proven ineffective in keeping up with the rapid growth and evolving nature of ransomware. Therefore, there is an urgent need for robust and effective methods to detect and mitigate these attacks.

This work proposes a novel ransomware detection approach based on behavioral analysis. An ensemble learning model is employed, where multiple feed-forward neural network base learners are trained on different classes of behavior using a combination of bootstrapping and feature subsetting techniques. Each base learner focuses on a specific class of features, such as API calls, file operations, and registry key operations. The aggregation step involves utilizing stacking techniques to create a final model trained on the predictions of the base learners. This combination of bootstrapping and stacking enhances the robustness and predictive power of the overall model. By combining the strengths of the base learners through ensemble learning, this approach aims to enhance accuracy and provide a comprehensive understanding of ransomware behaviors.

To assess the efficiency of the proposed model, it is compared against established machine learning algorithms such as logistic regression, support vector machines (SVM), and random forest, as well as state-of-the-art approaches. The results demonstrate that the proposed model outperforms other approaches in terms of accuracy, precision, recall and F1-score, effectively identifying ransomware outbreaks while reducing false positives. This research significantly advances ransomware detection methods and addresses the critical need for robust security measures in the ever-changing landscape of ransomware threats.

This thesis is organized as follows:

**Chapter 1** provides an overview of ransomware, including its evolution over the years. It explores the anatomy of ransomware attacks, delving into the various attack vectors employed by cybercriminals. This chapter sets the foundation for understanding the context and challenges associated with ransomware detection.

**Chapter 2** offers an overview of machine learning and deep learning techniques, as well as a review of previous studies in the field of ransomware detection. It establishes the theoretical framework necessary for developing the proposed model.

**Chapter 3** details the design and implementation of the ensemble classifier, presenting the model's architecture and training process. Furthermore, a comparative evaluation against traditional machine learning algorithms is conducted to assess the model's performance in terms of accuracy, precision, and recall. Real-world scenarios are also considered to evaluate the practical applicability of the model.

Finally, the thesis concludes with a discussion of the findings, their implications, and potential avenues for future research in the field of ransomware detection. The goal is to contribute to the ongoing efforts to combat ransomware attacks, enhance cybersecurity measures, and safeguard individuals and organizations from the devastating impacts of these threats.

# Chapter 1 State of the art

## .1 Introduction to ransomware

In recent years, we have witnessed a dramatic growth in the number of cyber attacks. Ransomware, in particular has become a global concern, posing challenges for individuals, businesses, organizations in almost every sector, especially for unicorn companies and governments, which are the big target for cybercriminals. Consequently, these entities have had to spend millions of dollars to strengthen their systems and protect their data (See Figure 1.2). There are various types of ransomware, each with their own unique characteristics and methods of attack. As a result, researchers have been doing a lot of work in this area, aiming to better understand and deal with this danger (See Figure 1.1) [37].

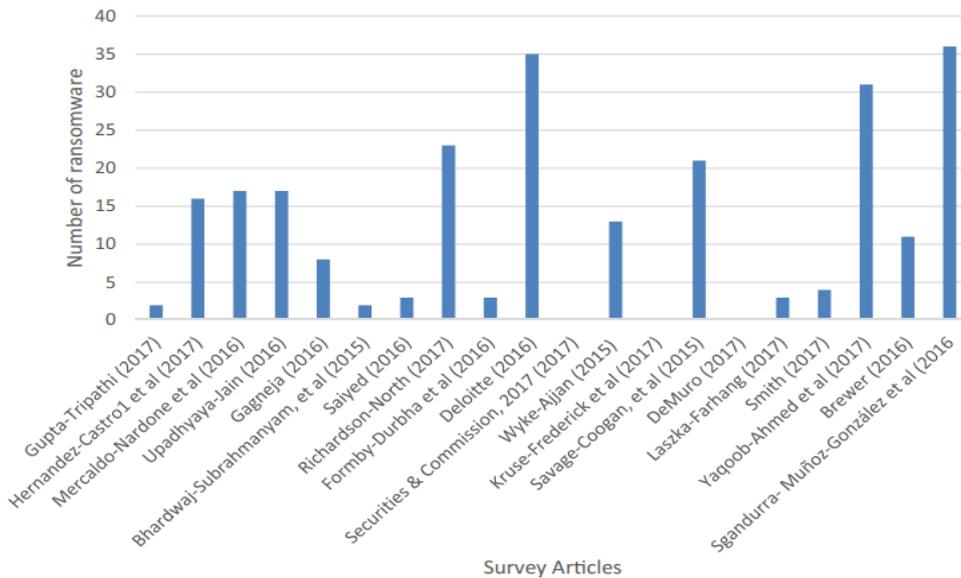


Figure 1.1 : Number of ransomware covered per survey article [83].

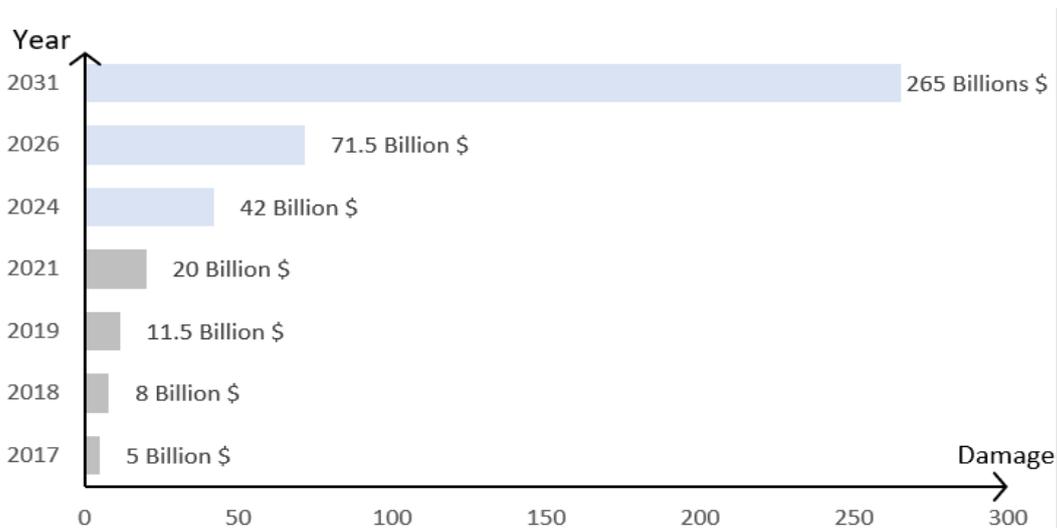


Figure 1.2 : Ransomware damage over the recent year.

## .1.1 Ransomware definition

Ransomware is a particular class of malware (malicious software) developed by cybercriminals to infiltrate a system, encrypt the victim's files and prevent or limit users from accessing their device, system and files. This software operates silently in the background, remaining undetected until it has achieved its objective, which is blackmailing the victim until the ransom is paid or risking public exposure of their personal data. The ransom claims usually come in the form of a message that appears once the files have been encrypted. Typically, the ransom is paid in Cryptocurrency formats such as Bitcoin and Ethereum.

## .1.2 Difference between ransomware and malware

### .1.2.1 Malware

Malware and ransomware are often used interchangeably, which is not correct. Ransomware is a class of malware and not all malware falls under the category of ransomware (See Table 1.1). Malware refers to any malicious software designed to harm or exploit computer systems or network. It can include viruses, worms, trojans, ransomware, spyware, adware [14][20][21][22]

### .1.2.2 Differences between malware and ransomware

Criteria	Malware	Ransomware
<b>Purpose</b>	Any malicious code designed to do a variety of actions such as damaging files and stealing bank account information.	Ransomware specifically focuses on encrypting or blocking access to a victims data or system until a ransom is paid.
<b>Target</b>	Malware can target both individuals and organizations, depending on its purpose. Some types of malware like adware might target individual users more.	Ransomware usually targets organizations more than individuals, as organizations are often more likely to pay larger ransoms.
<b>Symptoms</b>	Slower computer performance, unexpected crashes, unusual network traffic, unexpected pop-up messages, new unwanted programs...etc.	Unable to open files, seeing ransom messages on your screen, and finding that file names or extensions have been changed.
<b>Damage</b>	The damage caused by malware can vary widely, depending on its type and purpose. Some might slow down your computer, others might steal sensitive information or damage system files.	The damage caused by ransomware is more specific. It prevents access to files or systems, which can stop business operations, cause data loss, and other significant disruptions.
<b>Distribution (propagation)</b>	Malware can be distributed through a variety of methods: email attachments, infected websites, malicious downloads, USB drives.	Ransomware is often distributed through phishing emails, RDP Attacks or exploit kits that take advantage of security holes in a system

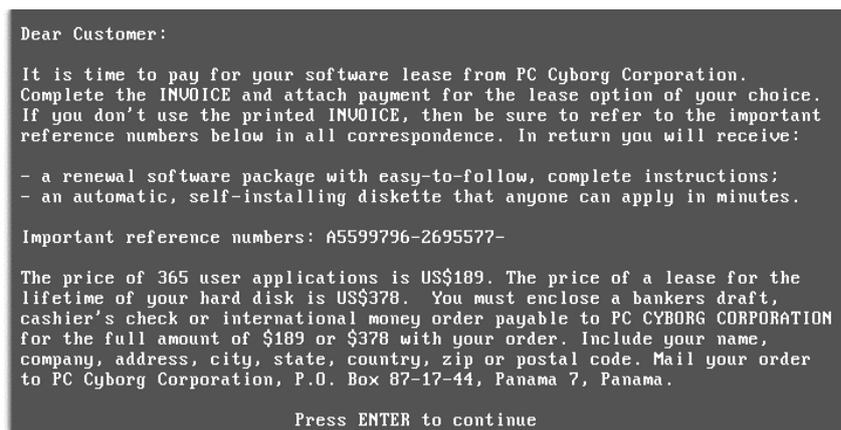
**Table 1.1** : Differences between malware and ransomware [14].

## **.1.3 Evolution of ransomware**

Ransomware has significantly evolved over the years in many aspects including the complexity of the threats, vectors of attack used and new tactics employed. Each year it becomes more sophisticated and harder to detect, we organize this in three important periods:

### **.1.3.1 The emergence of ransomware: A 1989-2006 Analysis**

The AIDS Trojan (also known as PC Cyborg) was the first ransomware attack in history; it emerged in December 1989 by Joseph Popp which was a biologist. He sent 20000 infected floppy disks labeled as AIDS research to researchers globally. When opened, the virus encrypted victims' files using simple cryptography. To recover access, victims had to pay 189\$ to a PO box in Panama. His creation paved the way for the ransomware industry. This attack highlighted the potential for cybercriminals to extort money through file encryption [1][2][9].



```
Dear Customer:

It is time to pay for your software lease from PC Cyborg Corporation.
Complete the INVOICE and attach payment for the lease option of your choice.
If you don't use the printed INVOICE, then be sure to refer to the important
reference numbers below in all correspondence. In return you will receive:

- a renewal software package with easy-to-follow, complete instructions;
- an automatic, self-installing diskette that anyone can apply in minutes.

Important reference numbers: A5599796-2695577-

The price of 365 user applications is US$189. The price of a lease for the
lifetime of your hard disk is US$378. You must enclose a bankers draft,
cashier's check or international money order payable to PC CYBORG CORPORATION
for the full amount of $189 or $378 with your order. Include your name,
company, address, city, state, country, zip or postal code. Mail your order
to PC Cyborg Corporation, P.O. Box 87-17-44, Panama 7, Panama.

Press ENTER to continue
```

**Figure 1.3** : AIDS trojan” ransomware note [1].

After a 15-year of silence ransomware reemerged again, GPCode and Archievus marked the beginning of the internet era for ransomware. GPCode was spread via malicious email encrypted victims files and demanding between 20 and 70 \$. Then in 2006 we saw Archievus which was the first ransomware to use a 1024-bit RSA encryption key which is tough to crack. It spreads through spam emails and harmful web addresses targeting “My Documents” folder; people who fell victim are obliged to purchase something from an online store to get a password which unlocks their folder [2][9][67].

### **.1.3.2 The Changing Face of Ransomware: A 2010-2019 Analysis**

In the second decade of the twenty-first century, we have seen various ransomware strains appearing characterized by their strong encryption algorithms and the introduction of cryptocurrencies as a mean of payment. We will cite some of them based on the significance and the financial impact (some of them are included in our dataset).

#### **2010-2014**

In 2010 WinLock gained notoriety as the first locker ransomware to capture public attention. It entered users systems through a malicious website and displayed inappropriate pictures. Upon infection, victims were directed to send 10\$. The cybercriminals gained about 16 million\$ [1][2].

Then Reveton appeared in 2012, which is also known as police ransomware. It was a type of financial ransomware that targeted Windows, Mac and mobile OS. It was distributed through drive-by-download attacks and presented victims with a fake alert accused the victims of committing a crime such as downloading pirated software and threatened by jail. This false message disguising to be as a message from law enforcement. The payments were in bitcoin. The cybercriminals made about 915000\$ from ransom payment [1][2][3][9].

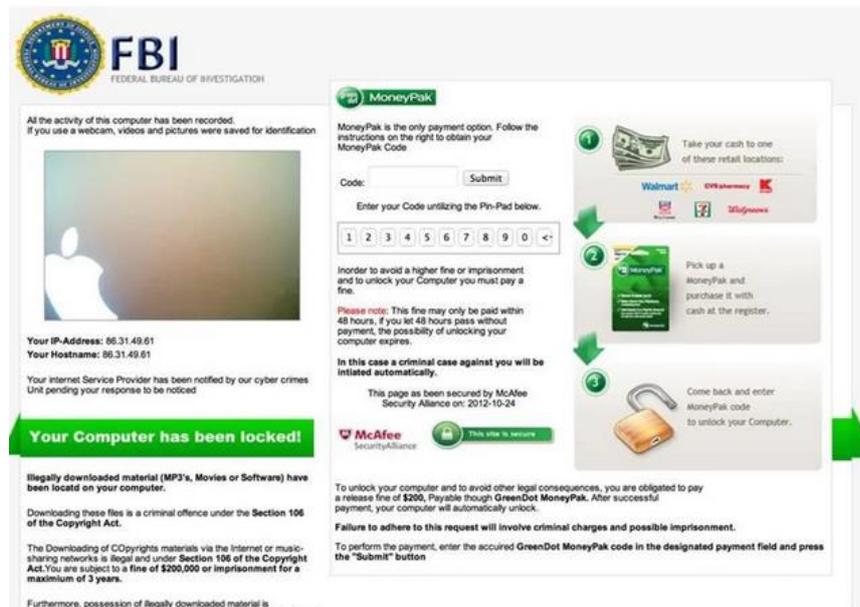


Figure 1.4 : Reveton ransom note [28].

Shortly after, in 2013, cryptolocker emerged as a highly sophisticated ransomware and with clever tactic. It locks the system and then encrypts both the system and any connected drives using a 2048-bit RSA encryption key. Even if a victim paid the ransom, the lock will be removed but their access to the system would remain blocked due to the encryption. Cryptolocker spread through phishing email with malicious attachments disguising as FedEx and UPS tracking notices, as well as via file sharing sites and downloads. This ransomware managed infecting around 250,000 computers worldwide, including an entire police department. The cybercriminals gained \$27 million from ransom payments [1][2][4][5].

Then in April 2014, CryptoWall was discovered which is considered as the imitator of CryptoLocker. In only six months, it infected 635000 systems and collected over \$1.1 million in ransom payments. It was distributed through exploits kits and via emails with zip attachments where the ransomware is hidden as pdf files [2][6][9]. In the same year, we saw the first ransomware target android devices which was simpleLocker ransomware, also known as SimpLocker. This malicious software encrypted various files like images, documents, videos using AES block chaining encryption algorithm with key length 128. Particularly, it also collected information such as device numbers, model numbers, manufacturers and even gained access to victims' cameras [2][7].

2015-2019

In this period, cybercrime took the step to a new dimension by the emergence of RaaS ransomware in 2015. In this period, we saw various attack vectors including malicious spam emails, exploit kits, and drive-by downloads. The financial loss reached 11 billion \$ in 2019, and we saw multiple variants of ransomware use new evasion tactics such as executing series of pre-attack API Calls [1][10][11].

We will cite some of them in brief, for instance in 2016 we have witnessed an explosion of ransoms with notable examples including Ransom32, Locky, Petya, Jigsaw and Zcryptor. Ransom32 became the first JavaScript ransomware capable of infecting Windows, Linux, and Mac OS. Zcryptor brought about a new concept of a cryptoworm, which had the ability to duplicate itself across networks and external devices.

The year 2017 witnessed the emergence of the most notorious threats in ransomware that made headlines in 2017 which is **WannaCry** (or WannaCrypt). This malicious software spread across the globe like wildfire. 150 countries were infected with about 5 million devices including UK national Health Service, FedEx, Honda and Boeing.

It used AES encryption each file with different key, it exploited a vulnerability in Windows OS, specifically targeting the component responsible for facilitating file sharing between computers [1][2][8][12][40][68].

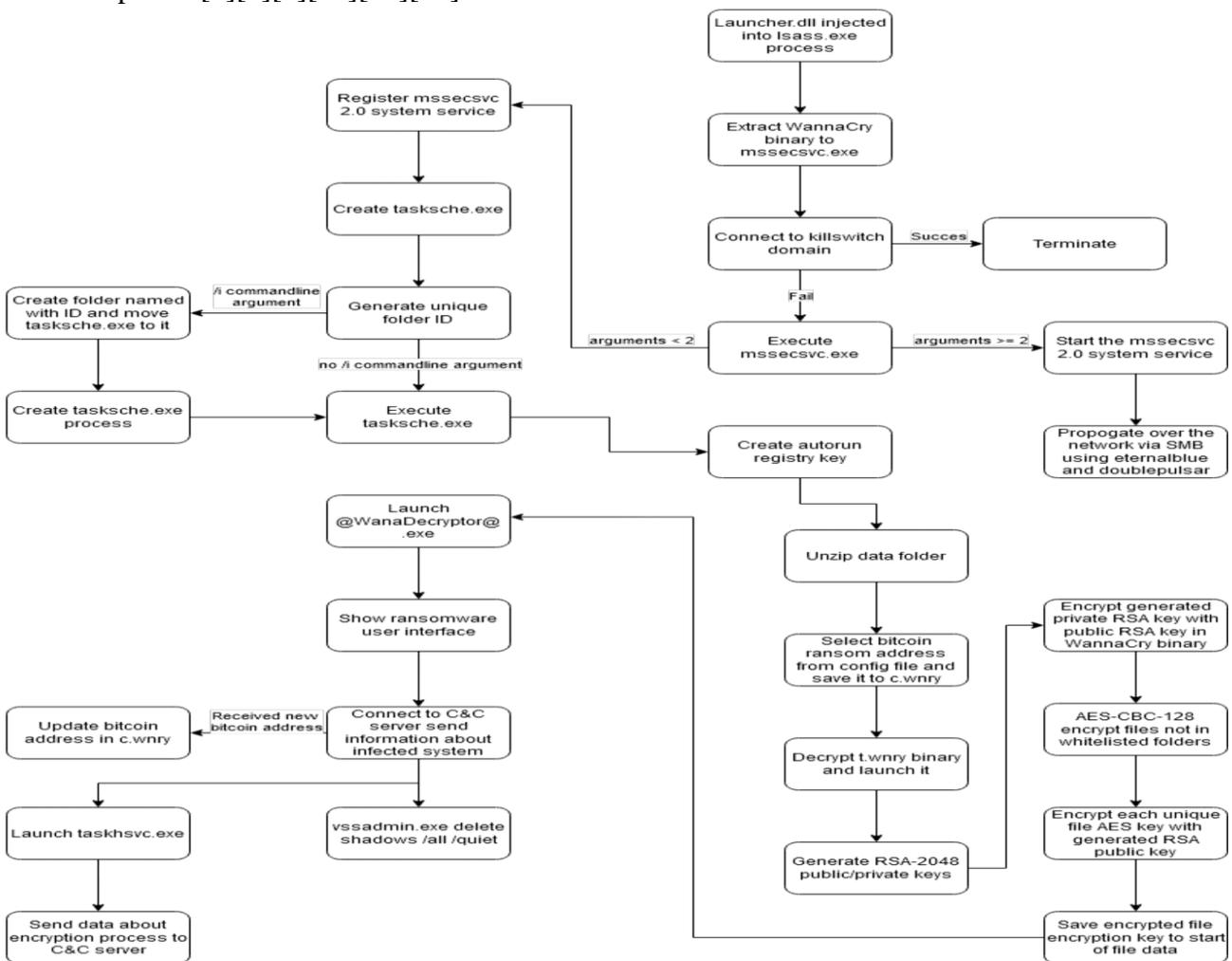
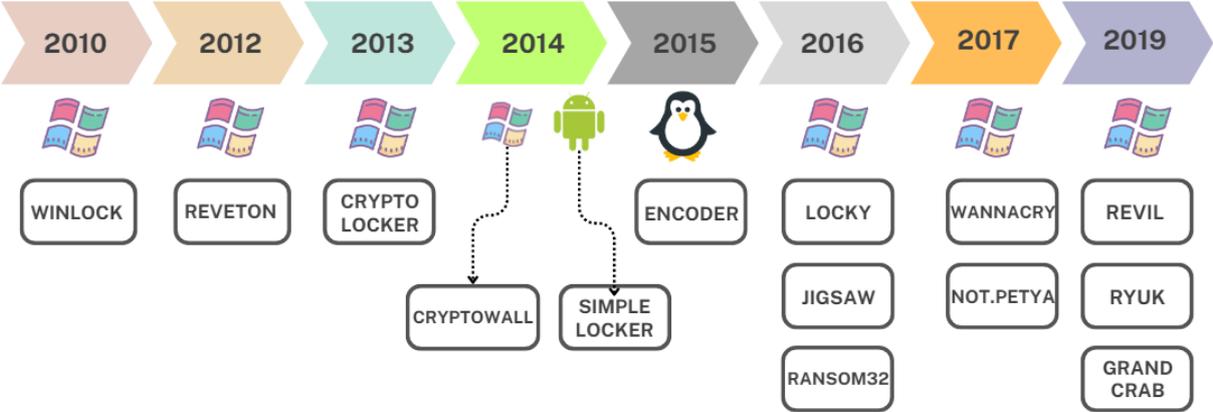


Figure 1.5 : WannaCry execution flowchart [10].

Another ransomware called **Ryuk** encrypted network drives, deleted backups, and made it hard to recover files. It came through fake emails with harmful documents. Ryuk targeted different organizations, including healthcare, government, and schools, and caused notable incidents like the attack on Tribune Publishing Company in December 2018.[40]

In 2019, notable double extortion ransomware strains appeared, each with distinct characteristics and targets such as REvil (or Sodin), maze, all of them employed various methods, such as exploiting vulnerabilities and phishing, RDP attacks, and exploit kits.



**Figure 1.6 :** Evolution of major ransomware families from 2010 to 2019.

**.1.3.3 Recent ransomware (2020-2023)**

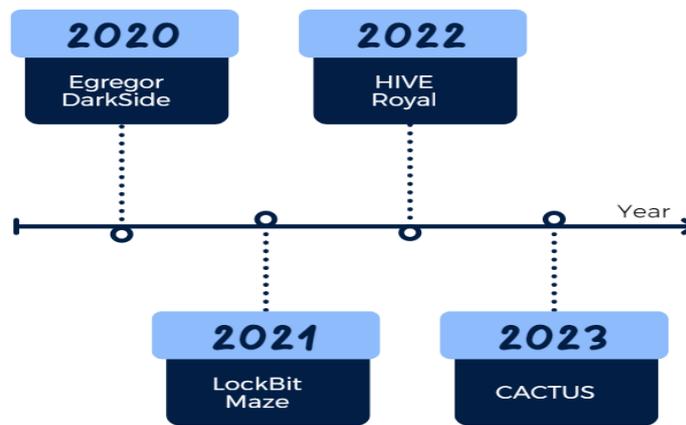
In recent years, there has been a significant increase of ransomware attacks where they become more persistent, complex and costly than ever. This era has witnessed the emergence of a new type of ransomware called double extortion, which surpasses its predecessors in terms of effectiveness and sophistication. This new variant has adopted a strategy known as practicing big game hunting, targeting larger organizations over individual users for higher ransom payment.

We will highlight the 3 important ransomwares strains of this period namely Cactus, Egregor, and DarkSide.

In September 2020, Egregor ransomware emerged as a double extortion strain, publicly shaming victims, and demanding ransoms. Ironically, upon payment, attackers offered victims advice on network protection [2].

In August 2020, DarkSide appears focused on high-profile targets, employing double extortion and stealth tactics. Notably, it carried out the May 2021 Colonial Pipeline attack. Pressured by the U.S. government, DarkSide announced its operational suspension in May 2021, yet influenced new ransomware groups like BlackMatter [68].

In March 2023, Cactus ransomware surfaced, exploiting VPN vulnerabilities to infiltrate networks and using self-encryption to evade detection. It used unique AES and RSA keys for decryption and file encryption, respectively, and distinct file extensions during the encryption process. Cactus utilized Cobalt Strike malware and Chisel for command-and-control, proving the growing complexity of ransomware. The escalating sophistication of these ransomware underscores the need for strong security measures and consistent system updates [17].



**Figure 1.7 :** New ransomware strains from 2020 to Quarter1 of 2023.

## .1.4 Anatomy of ransomware attacks

### .1.4.1 Ransomware attack vectors

Ransomware attacks can occur through various vectors, which are the methods or pathways through which the ransomware infiltrates a system. Here are some common ransomware attack vectors:

**a. Phishing email:**

One of the most widespread methods is through malicious email attachments. Attackers send spam emails posing as legitimate entities or containing infected attachments. When users open the attachment, the ransomware is installed on their system [8][19][23][26][40][61].

**b. Drive-by downloads attacks:**

Ransomware can be distributed through compromised websites or deceptive ads. When a user visits an infected website or clicks on a malicious advertisement, the ransomware is automatically downloaded and executed on their device without their knowledge [40].

**c. Exploit kits:**

Attackers take advantage of software vulnerabilities in most used software like web browsers, plugins like adobe flash player or operating systems. They utilize exploit kits, which automate the process of identifying and exploiting these vulnerabilities. By deploying these kits, hackers can deliver ransomware to unsuspecting users through compromised websites, infecting their devices without their knowledge or permission [26][61].

**d. Remote Desktop Protocol (RDP) attacks:**

RDP is a protocol developed by Microsoft that allows users to connect to and control a remote computer through a network connection. Unfortunately, compromised RDP has been a common attack vector for ransomware. When RDP is compromised, attackers exploit weak RDP credentials to gain unauthorized access to a system; once they have gained entry, they can install ransomware and take control of the victim's files and systems [25][26].

**e. Malicious downloads:**

Users may unknowingly download and install infected software or files from untrustworthy sources, including torrent sites, peer-to-peer networks. These downloads can contain ransomware that gets executed upon installation.

**f. USB and removable media:**

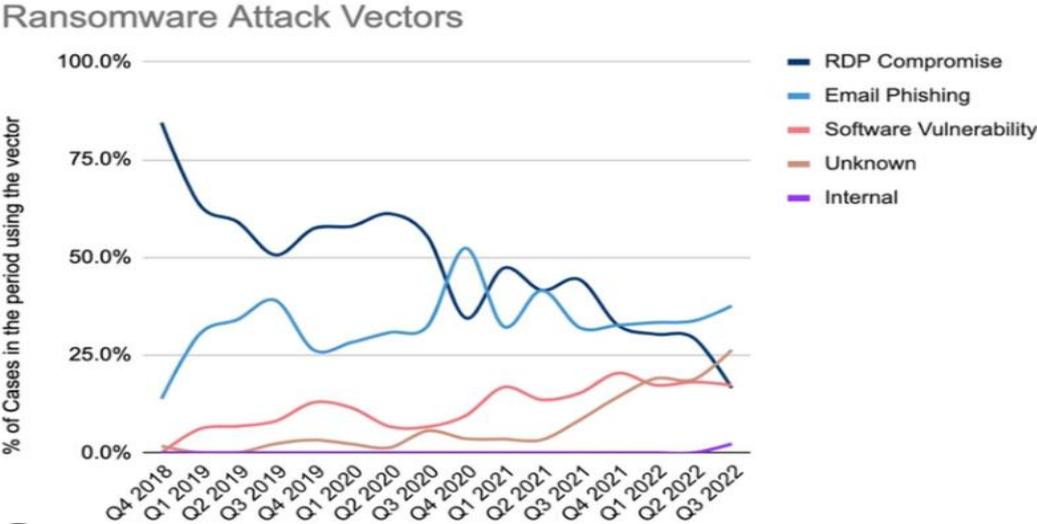
Ransomware can spread through infected USB drives or other removable media. When users connect these devices to their computers, the ransomware is automatically executed and starts infecting files.

**g. Malvertising:**

Cybercriminals use online ads to distribute ransomware. Malicious advertisements are designed to look legitimate and may redirect users to compromised websites or initiate downloads of ransomware onto their systems [23][27][28].

**h. Social engineering:**

Social engineering is one of the most successful ransomware attack methods, they take advantage of human weakness by tricking users into executing ransomware themselves. This can involve deceiving users through phone calls, pretend to be tech support or pressing them into running scripts or software [23][24].



**Figure 1.8 :** Ransomware Attack vectors between 2018 and 2022 [78].

### **.1.4.2 The Six Stages of Ransomware Attacks**

Ransomware attacks vary, but they generally follow a pattern which consists of six major stages:

#### **1. Campaign**

Ransomware attacks start with a campaign which is the method that a hacker employed to deliver an attack using one of ransomware attack vectors. The campaign is essentially the initial step in delivering the ransomware payload to the targeted victims [15][16].

#### **2. Infection**

This phase begins when the ransomware is executed on the system, and will spread across the system [15][16].

#### **3. Staging**

The ransomware embeds itself into the system, establishing persistence, and communicates with a command-and-control server [15][16].

#### **4. Scanning**

The malware scans the network to identify files to encrypt, including data stored in the cloud [15][16].

#### **5. Encryption**

After the malware completes its analysis and inventory, it initiates an encryption process. Local files receive near-immediate encryption. Then, the malware moves to shared files on the network. Data on the network is copied locally, encrypted, then uploaded back to the share so that it replaces the original document [15][16].

#### **6. Remuneration**

Hackers frequently demand payment from network by sending a ransom note to their devices. They send also instructions on how to pay the ransom, which is typically in Bitcoin. In certain cases, hackers may increase the ransom amount over time to pressure network administrators. Some of this cybercriminals even provide "customer service" lines to help their victims.

If the victim decides to pay the ransom, we will proceed to the second stage, which involves recovery and cleanup from the ransomware.

- **Recovery:**

The victim pays the ransom, which may or may not result in restoration of data and files.

- **Cleanup:**

The organization's cybersecurity team or consultant analyzes the attack, ensures that malware has been removed, and takes steps to prevent further attacks

### .1.4.3 How Ransomware keeps your digital life as hostage

Developed countries and big companies take ransomware very seriously. They understand the serious risks it poses.

**Table 1.2** explains why they are considered a highly dangerous. We provide a description of various features that highlights the power and advanced strategies and techniques employed by ransomwares in today's landscape:

Features	Description
<b>Inviolable encryption</b>	Ransomware employs strong encryption techniques that are near impossible to decode using decryption tools. This means once files are encrypted, it's extremely difficult to regain access without the encryption key.
<b>Multiple encryption tools</b>	The encryption methods used by ransomware are versatile and powerful enough to encrypt all file types (audio, images, document and even video)
<b>Obfuscatory extensions</b>	Once the victim's data is captured by ransomware, it obfuscates the file names and extensions to confuse the victim, in addition, it complicates efforts to recover or understand the extent of the encrypted data.
<b>Impenetrable control message</b>	The ransomware captures all the data of victim system and encrypts. Once all the data of victim is encrypted, it presents a ransom demand via a control message. This message, often in the victim's native language, is impossible to bypass until the ransom is paid.
<b>Untraceable payment mode</b>	Ransomware claims are typically untraceable because we use cryptocurrencies as ransom like Bitcoin or Ethereum. This helps the attackers evade tracking and lawsuit.
<b>Time bound payment</b>	The ransom payment has to be done within the time limit mentioned in the control message. Failure to meet this deadline can lead to an increase in the ransom amount or permanent data loss.
<b>Extendable threats</b>	If the victim's system is connected in a network, ransomware can potentially extend its reach to all other connected systems, increasing the size of the attack.
<b>Complex set of exfiltration techniques</b>	Ransomware often employs sophisticated methods to extract passwords, usernames, mail and ID and other sensitive data from the victim's system, increasing the potential damage.
<b>Geographical targets</b>	In some scenarios, some ransomware attacks specifically target systems in certain geographical areas, indicating a level of strategic planning in their execution.

**Table 1.2** : Vitality of ransomware [77].

(Defining **vitality** as the ability of ransomware to persist, propagate, and resist.)

#### **.1.4.4 Ransomware Component**

Ransomware attack is typically executed through a series of components, each playing a crucial role in the attack lifecycle.

##### **a. Dropper:**

The dropper is responsible for delivering the payload to the victim's system. It could be a malicious email attachment, a drive-by download from a compromised website, or a malicious app.

##### **b. Exploit Kit:**

These are tools used to exploit software vulnerabilities in order to install the ransomware. The victim might not even realize that their system has been compromised.

##### **c. Payload:**

This is the actual ransomware program that carries out the encryption process. It contains the encryption algorithms and other functionalities like self-propagation.

##### **d. Command & Control (C&C) Server:**

The C&C server is a remote server controlled by the attacker. The ransomware communicates with this server to exchange information about the infected system, receive encryption keys, and relay payment confirmations [8].

##### **e. Encryption algorithm:**

Ransomware typically employs two types of encryption: symmetric and asymmetric. Symmetric encryption uses a single key to both encrypt and decrypt information such as AES algorithm, or asymmetric encryption like RSA algorithm, which involves a pair of keys a public key for encryption and a private key for decryption. The private key is stored on the C&C server and is only provided to the victim upon payment.

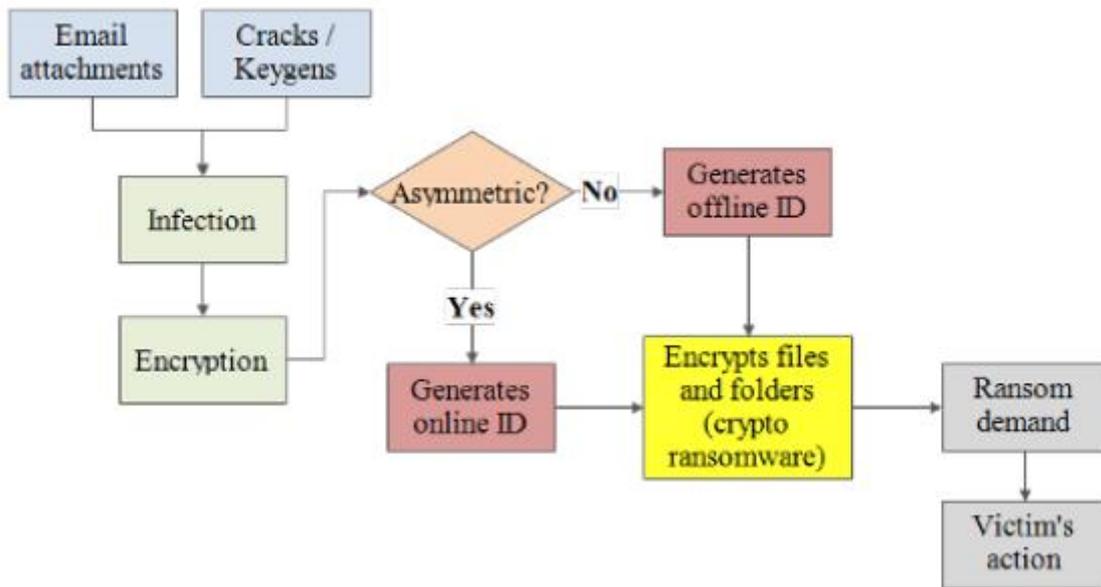
Recent ransomware has started using a combination of symmetric and asymmetric algorithms which is called hybrid encryption. This approach increases the complexity of the decryption process and takes advantage of the benefits provided by both encryption which is speed and security [8].

##### **f. Ransom Note:**

This is a message from the attacker, usually a pop-up or a text file, explaining what happened to the victim's files and providing instructions on how to pay the ransom.

##### **g. Decryption Tool:**

If the ransom is paid, the attackers provide a tool or key to decrypt the encrypted files. However, there is no guarantee that the attackers will provide a working decryption tool even if the ransom is paid.



**Figure 1.9** : Djvu Ransomware's sequence of operations [28].

Classic ransomware components still exist today, and new elements have appeared making ransomware more challenging to combat. These advancements include:

#### **AI Techniques:**

Some ransomware now use AI to improve its effectiveness. For example, AI can be used to analyze the victim's behavior and tailor the attack accordingly, making it more likely to succeed [69][70].

#### **Polymorphic Code:**

This allows the ransomware to change its code each time it propagates, making it harder for antivirus software to detect [71][72][73].

#### **Worm Capabilities:**

Some modern ransomware can spread across networks, infecting multiple machines without user intervention [8].

#### **Data Exfiltration:**

Before encrypting the victim's files, some ransomware will first exfiltrate the data. This gives the attacker additional advantage as they can threaten to release the data publicly if the ransom is not paid.

## .2 Types of ransomwares

Ransomware can be classified in various ways such as:

- Classification based on target environment or the specific systems they are designed to attack,
- Classification based on infection (propagation methods),
- Classification based on communication,
- Classification based on the primary mechanism of attack.

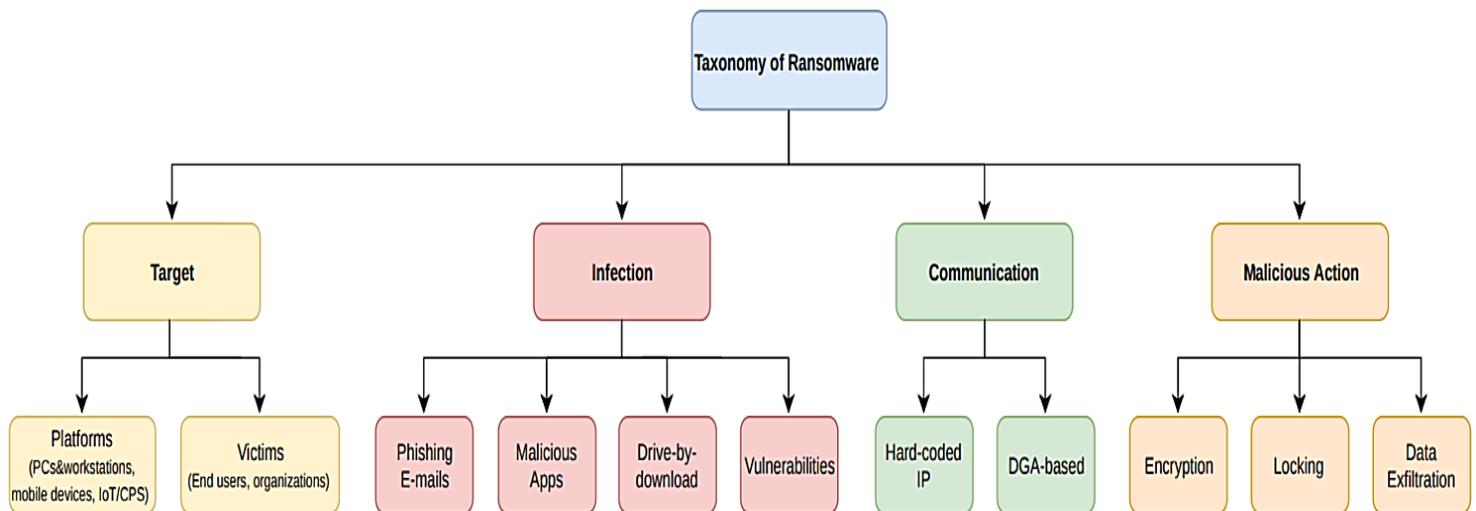


Figure 1.10 : Taxonomy of ransomware [8].

Here we will focus on the primary mechanism of attack.

There are two major categories of ransomware which are **Encryptors** and **Lockers** [19].

### .2.1 Locker

As the name indicates, this type of ransomware locks the victim's system instead of encrypting files. It aims to prevent the victims from accessing their own system; it will take control of the whole system and display a notification demanding a ransom payment. The message may contain threats of file deletion if the ransom is not paid. Some well-known Locker ransomware families include WinLock, Reveton, Locky, Jigsaw [2][8][19][28].

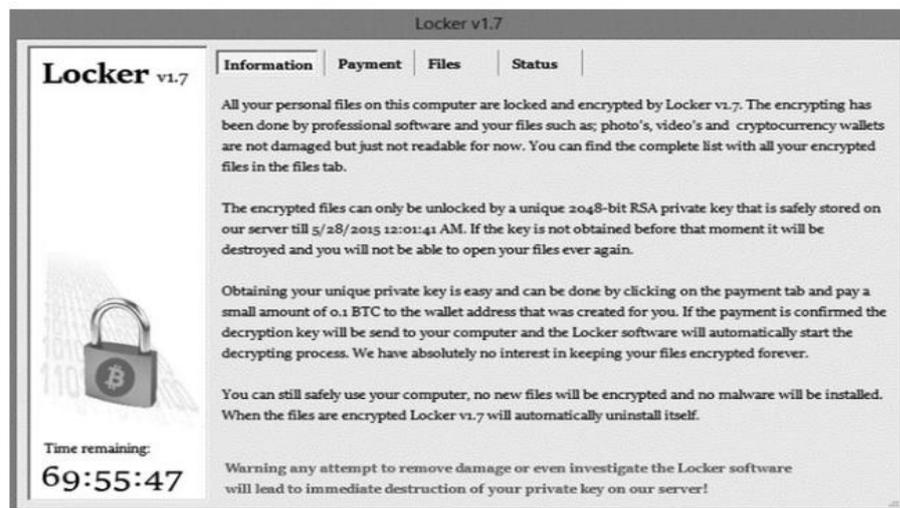


Figure 1.11: Locker Ransom note [79].

### .2.1.1 How Locker behaves inside a system:

Infection	Actions within a system	Communication with C&C server	Ransom demands
Locker ransomware typically infects a system through phishing emails, malicious downloads, or exploit kits. The user unknowingly executes the malware, which then takes over the system	Once inside, the ransomware disables key system functions, effectively locking the user out. It may also disable security software to prevent removal.	The ransomware establishes communication with a command and control (C&C) server to relay information about the infected system and receive further instructions.	The user is presented with a ransom note, typically demanding payment in cryptocurrency. The note threatens that the system will remain locked until payment is made.

**Table 1.3 :** How locker behave inside a system.

## .2.2 Encryptor

Also known as crypto ransomware, unlike locker this variety of ransomware uses encryption algorithms such as AES and RSA to encrypt files or in some cases deletes or overwrites the original files and demands a ransom payment for decryption key. Some of the newer variant use a combination of symmetric and asymmetric algorithms to enhance the complexity of the decryption process. Some famous Encryptor ransomware families include WannaCry, GrandCrab, NotPetya, BadRabbit, Jigsaw [2][8][19][28][40].

### .2.2.1 How Encryptor behaves inside a system:

Infection	Actions within a system	Communication with C&C server	Encryption process	Ransom demands
Similar to Locker ransomware, Encryptors also spread through phishing emails, malicious downloads, or exploit kits.	Once inside, the ransomware scans the system for files to encrypt. It may target specific file types such as documents, images, or databases.	The ransomware communicates with the C&C server to send the encryption key and receive further instructions.	The ransomware uses strong encryption algorithms (like RSA or AES) to encrypt the files. The encryption key is generated and sent to the C&C server, ensuring only the attacker can decrypt the files.	The user is presented with a ransom note, demanding payment for the decryption key. Failure to pay often results in the key being destroyed, leaving the files permanently encrypted.

**Table 1.4 :** How Encryptor behave inside a system.

## .2.3 Leakware

Leakware, also known as Doxware, is a new variant of ransomware that steals sensitive or personal data from victims and threatens to publish this information in public if the victim does not pay the ransom. Unlike other types Leakwear does not necessarily encrypt files [19][29].

Leakware can be particularly damaging to organizations over individuals, as the public release of sensitive data can have negative effects. These could include damage to the company’s reputation, loss of customer trust and significant financial losses.

Infection	Actions within a system	Data exfiltration	Communication with C&C server	Ransom demands
Leakware also spreads through phishing emails, malicious downloads, or exploit kits.	Once inside, the ransomware scans the system for sensitive data to exfiltrate.	The ransomware sends the sensitive data to the C&C server.	The ransomware communicates with the C&C server to send the exfiltrated data and receive further instructions	The user is presented with a ransom note, threatening to publish the sensitive data unless payment is made. Failure to pay results in the data being published or sold.

Table 1.5 : How Leakware behave inside a system.

## .2.4 Other types of ransomware:

### .2.4.1 Double extortion

Double extortion ransomware is an evolution of traditional ransomware tactics, adding another layer of threat. In a double extortion attack, the attackers infiltrate a target’s network using advanced techniques like social engineering tactics, then they extract sensitive data before deploying ransomware to encrypt the victim's files. This stage is undetected, allowing attackers to quietly collect valuable information.

Shortly after, second stage begins, which involves the typical ransomware attack, encrypting the victim’s files and demanding a ransom in exchange for the decryption key. In case where the victim refuses to pay or attempts to restore their systems from backups, the attacker threatens to leak the stolen data online or sell it on the dark web.

The most famous one was Maze, which steals sensitive data and threaten to leak it on their "Maze News" site [2][31].

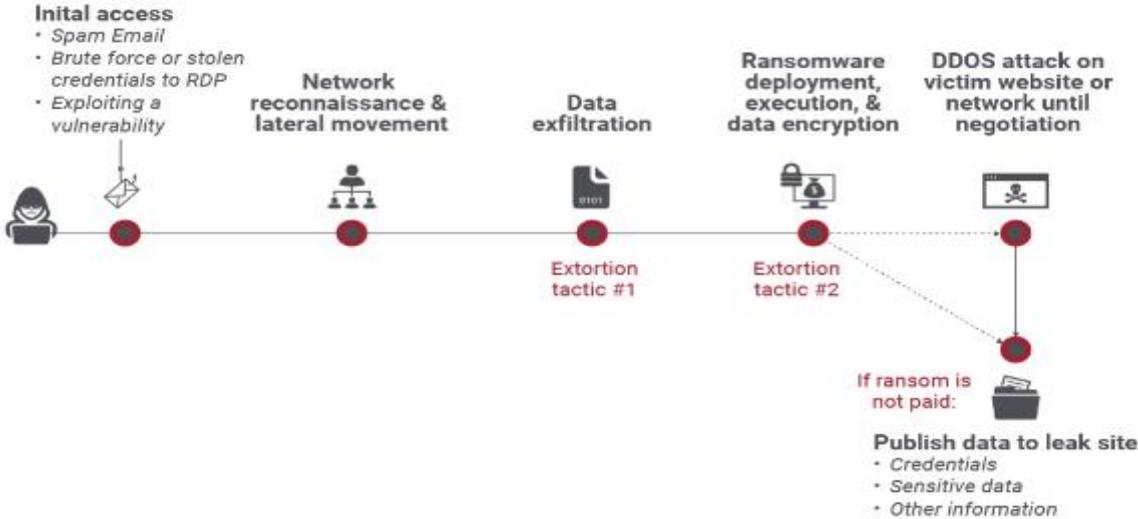


Figure 1.12 : How Double extortion ransomware works [31].

## **.2.4.2 Scareware**

Scareware is another type that usually uses social engineering tactics to trick users into thinking their computer is infected with ransomware by showing a notification message includes a warning, and they need to purchase a fake anti-virus software to fix the issue. Scareware can be distributed through email attachments, pop-up ads, or by exploiting vulnerabilities in software or operating systems. Some types of scareware lock the computer, while others simply flood the screen with pop-up alerts without actually damaging files [40].

## **.2.4.3 RaaS (ransomware as a service)**

Ransomware as a Service (RaaS) is a cybercriminal business model where ransomware technology is developed and either sold or leased to other criminals who execute the attacks. In this model, the creators of the ransomware charge a fee or a portion of the profits generated from the ransom attacks. This approach appeals to individuals who desire to partake in cybercrime but lack the necessary technical expertise, as they can effectively rent the ransomware and its infrastructure from a provider. RaaS operations usually offer purchasers a user-friendly interface to specify ransom amounts, payment deadlines, ransom messages, and more, enabling them to deploy the ransomware against their chosen targets. Each RaaS campaign utilizes its own distinct ransomware, which can vary in complexity but generally works by encrypting the victim's files and demanding a ransom, usually in cryptocurrency, in exchange for the decryption key [2][8][19][28][33][68].

# **.3 Fundamental techniques for analyzing ransomware**

## **.3.1 Static Analysis:**

Static analysis involves studying the malware code without running the executable. The key advantage of static analysis is that it can quickly provide high-level information about the malware, such as its potential behavior and threat level.

Static analysis includes techniques like signature-based detection, where the malware is compared to known malware signatures. Reverse engineering is also commonly used, where the malware code is decompiled to understand its structure and functionality. However, modern ransomware often employs obfuscation techniques, making static analysis challenging [8][63].

Static analysis typically involves techniques such as:

### **a. File Signature Analysis:**

This technique involves checking the malware file against a database of known malware signatures. If a match is found, it gives an immediate identification of the malware.

### **b. Code Disassembly:**

This involves using tools called disassemblers to convert the binary code of the malware into assembly language such as IDA Pro or Ghidra. The analyst can then inspect the code to understand what it does [8].

**c. Strings Analysis:**

This technique involves extracting human-readable strings from the malware binary, which can sometimes provide insights into its function or origin.

**d. Cryptographic Analysis:**

Since many malware authors use encryption or obfuscation to hide their code, cryptographic analysis can be used to detect and potentially break these methods.

**Advantage and limitation:**

Static analysis has the advantage of being safe (since the malware is not actually run) and can provide a quick overview of the malware's function. However, it also has limitations. For example, it can be time-consuming to understand complex malware fully, and it may not reveal the malware's full behavior, particularly if the malware uses anti-analysis techniques.

## **.3.2 Dynamic Analysis**

Dynamic analysis involves executing the malware in a controlled environment, typically a virtual machine or sandbox. By observing the malware's actions while it's running, analysts can gain an in-depth understanding of its behavior, including its encryption methods, communication channels, and data exfiltration tactics. It is the most used type of analysis in literature studies (See **Figure 1.15**) [63].

Dynamic analysis techniques may include:

**a. Behavioral Analysis:**

This involves running the malware in a controlled environment, such as a sandbox, and observing its behavior. This could include which files it modifies, which network connections it makes, and which system processes it interacts with.

**b. Debugging:**

This involves running the malware in a debugger, a special program that allows the analyst to control the execution of the malware step by step and observe its behavior in detail.

**c. Memory Analysis:**

Since many malware types operate in memory to evade detection, analyzing the system's memory can provide valuable insights into the malware's function.

**Advantage:**

Dynamic analysis has the advantage of revealing the malware's actual behavior rather than potential behavior, as seen in static analysis. However, sophisticated ransomware can often detect when it's being run in a virtual environment and modify its behavior to evade detection.

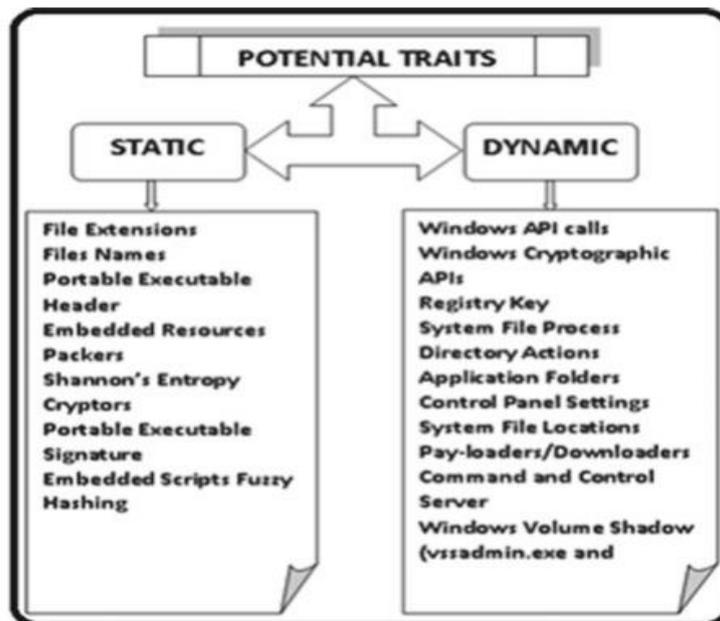


Figure 1.13 : Potential traits (ransomware dataset features) [77].

### .3.3 Hybrid analysis

Each method (static or dynamic analysis) has its strengths and weaknesses and is suited to different scenarios.

In practice, malware analysts often use a combination of both static and dynamic analysis. Static analysis can provide a quick overview and help direct the dynamic analysis. In contrast, dynamic analysis can reveal the malware's behavior in a real-world scenario, providing insights that may not be evident from the static analysis alone [8].

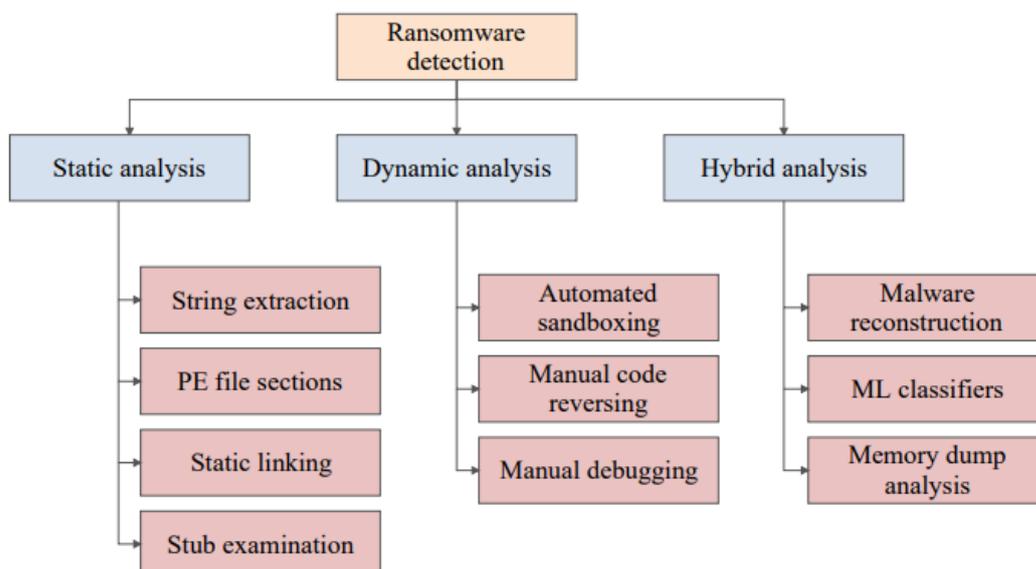


Figure 1.14 : Taxonomy of Ransomware detection techniques [8].

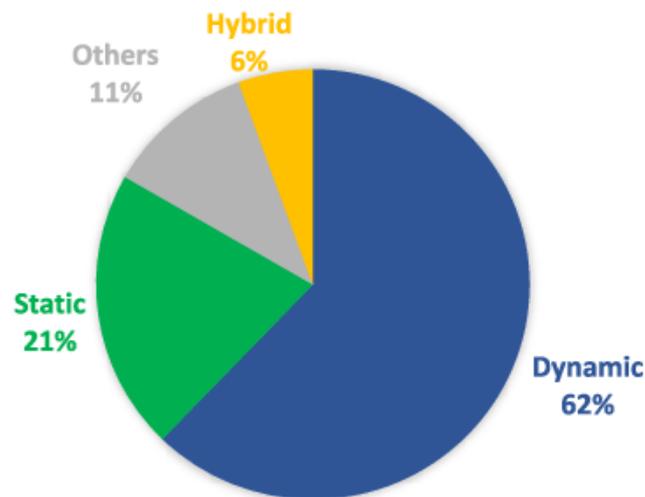


Figure 1.15 : Type of analysis distribution in Literature Studies [19].

## .4 Traditional ransomware detection

### .4.1 Signature-based detection methods:

Signature-based detection is one of the oldest methods for identifying malware. This method relies on identifying known patterns or signatures of ransomware in files and applications. It is effective against known ransomware variants but may not detect new or modified threats

Signature-based detection is the simplest way to identify the presence of malware on a system. Malware signatures include information like file hashes, the domain names and IP addresses of command and control infrastructure, and other indicators that can uniquely identify a malware sample. Signature-based detection systems store a library of these signatures and compare them to each file entering or running on a system to see if it is malware [63].

However, signature-based detection is growing less and less useful. It has significant limitations when dealing with ransomware [8].

#### **Evolving Signatures:**

Ransomware authors often modify their code to create new variants with different signatures. These variants can evade signature-based detection until their signatures are identified and added to the database.

#### **Zero-Day Attacks:**

Signature-based detection is ineffective against zero-day attacks, which exploit previously unknown vulnerabilities. Since these attacks use entirely new malware, their signatures are unknown.

## **.4.2 Behavior-based detection methods**

Behavioral detection is another option for detecting the presence of ransomware on a system. This approach monitors system activities and looks for unusual or suspicious behavior that may indicate a ransomware infection. It can detect new and unknown ransomware variants

Behavior-based detection algorithms can be designed to look for specific activities that are known to be malicious or to look for anomalous actions that differ from the norm.

Behavior-based ransomware detection takes advantage of the fact that ransomware has very unusual behavior. For example, ransomware's encryption stage requires the malware to open many files on the system, read their contents, and then overwrite them with an encrypted version. This behavior can help with ransomware detection if an anti-ransomware solution monitored file operations or encryption operations and alerted on this unusual behavior [8][63].

However, behavior-based detection also has limitations:

### **False Positives:**

Since legitimate software can sometimes exhibit behavior similar to malware, behavior-based detection can result in false positives.

### **Evasion Techniques:**

Sophisticated ransomware can employ various evasion techniques, such as operating slowly to avoid sudden spikes in CPU or disk usage, or remaining inactive until certain conditions are met

## **.4.3 Heuristic-based methods**

Signature based and behavior-based ransomware detection methods have some drawbacks. Hence, heuristic ransomware detection methods are proposed to overcome these disadvantages. Heuristic ransomware detection methods use machine learning techniques to learn the behavior of an executable file [66]. A system using heuristic methods keeps a close eye on three main areas:

### **a. File behavior**

When we talk about 'file behavior', we are referring to the actions a file takes or the activities it engages in when executed. In a normal situation, a file (like a word document or an application) would perform its intended function, such as opening text for reading or launching an application for use. However, a file infected with ransomware might behave differently. It might try to alter or encrypt other files, prevent certain applications from running, or even attempt to spread itself across your network. Heuristic detection observes these behaviors and flags any actions that seem unusual or suspicious. It is like a security guard keeping a watchful eye on a crowd, ready to spot anyone who's acting out of line.

### **b. System monitoring**

This is similar to installing security cameras throughout a building to keep tabs on what's happening. The system in this case is your computer or network. Under normal conditions, your system will carry out expected operations. However, ransomware or other malware can cause anomalous activities, like a sudden spike in data usage, repeated failed login attempts,

or unusual modifications to system settings. System monitoring in heuristic detection is about tracking these activities, looking for patterns that might indicate a cyberattack.

If a particular action or sequence of actions deviates significantly from the norm, the system flags it for further investigation.

**c. User behavior**

Every user has a specific pattern of computer use. For example, you might check your emails first thing in the morning, then work on a document, followed by a quick check of social media sites. Over time, these patterns form a 'behavioral profile.' Now, imagine if one day your computer starts sending out hundreds of emails per minute or tries to access confidential files in the middle of the night - activities that do not fit your normal profile. These anomalies can be signs of a potential cyber threat, such as ransomware. User behavior analysis in heuristic detection involves creating a baseline of normal behavior and flagging any significant deviations from this baseline.

## **.5 Tradition solutions**

**a. Ensure data backup:**

To safeguard against ransomware, it's recommended to have a plan for backup and recovering your data. Ransomware locks up your data making it impossible to access. Regularly backing up your data is one of the best ways to protect against ransomware. If your data gets encrypted, you can simply restore it from the backup copies. Just make sure to store the backups securely and keep them disconnected from the network or the computer being backed up. This precaution is important because ransomware can also encrypt backups that are connected to the network [9][37][40][51][59].

**b. Update software:**

Regularly updating software with the latest patches and updates is crucial to prevent ransomware attacks, as ransomware often takes advantage of known vulnerabilities in applications and operating systems. By staying up-to-date with software updates, you can significantly reduce the risk of falling victim to ransomware.

**c. Use Anti-malware Solution:**

Anti-Malware solutions can help protect against ransomware by detecting known malware signatures, analyzing suspicious behavior, and providing real-time protection to prevent malware from spreading or locking up files. They're regularly updated to guard against new threats, and if an infection occurs, they may offer tools to remove the ransomware and restore the system. However, these solutions are not foolproof, and it's crucial to practice good cybersecurity habits in addition to using anti-virus software [9][40].

**d. Educating yourself & your employees:**

Educating Employees about the new techniques used by cybercriminals and discussing the preventive measures organizing the security awareness pieces of training. And giving them basic cybersecurity training [40].

**e. Filter the emails:**

To prevent phishing emails, we should use email filters that can be configured to block any untrusted sources [40].

Unfortunately, even though there are several preventing measures, but they are still not sufficient to prevent or mitigate ransomware attacks.

In recent years, new and clever tactics have emerged such as the attack loop that can easily bypass backup solution. This method involves infecting a system with ransomware using traditional means but delaying its activation for several months. This allows the ransomware to become embedded in the system's backups. When the ransomware finally activates, it demands a ransom for decryption. If you attempt to restore the system using a previously assumed clean backup, the ransomware persists, leading to a reactivation of the attack. This creates a loop where you continuously find yourself back at the step of facing the ransom demand.

For instance, the KeRanger ransomware specifically targeted Mac computers and included a feature to encrypt backups created with TimeMachine, an OS X backup software

Furthermore, implementing anti malware solution is not effective 100%, it may fail to detect new ransomware strains that are designed to evade anti malware software. In addition, it can also slow down your computer

Additionally, even if we educate employees, the action of a single employee who does not follow instructions or does not apply best practices is sufficient for a professional hacker to infiltrate the system [51][52][59].

## **.6 Conclusion**

In this chapter, we covered an overview of ransomware, its evolution and various types. We have discussed the primary mechanisms of ransomware attacks and the emergence of double extortion and leakware variants. We have explored different techniques for analyzing ransomware, traditional ransomware detection methods and their limitations.

The chapter concludes by emphasizing the ongoing challenges in mitigating ransomware risks and the necessity of innovative defensive strategies, which have become crucial.

# ***Chapter 2 Machine Learning and Deep Learning***

## **.1 Introduction**

The previous chapter emphasized the insufficiency of traditional preventive measures in the face of the increasing complexity and sophistication of ransomware. Therefore, it is imperative to explore advanced and innovative solutions. In this chapter aims to delve into various machine learning algorithms then we will explore widely used deep learning models in order to enhance the detection of ransomware attacks. Through the utilization of these advanced techniques, our goal is to improve our ability to identify and respond to new threats.

## **.2 Machine Learning**

Machine learning is a branch of artificial intelligence that focuses on the development of algorithms and statistical models enabling computer systems to perform specific tasks and make predictions based on data. These systems learn and get better from real world experiences rather than relying on programming. The goal of machine learning is to create models that can analyze patterns and relationships within large datasets [32].

### **.2.1 Overview of Machine learning application in cybersecurity**

Machine learning has revolutionized the field of cybersecurity by providing advanced tools to detect and respond to cyber threats. Traditional methods often rely on predefined rules and patterns to identify malicious activities, but these approaches can be limited in their effectiveness. Machine learning, on the other hand, uses algorithms that can learn from vast amounts of data to recognize patterns and anomalies that may indicate cyber-attacks.

Machine learning algorithms can analyze large datasets of malware samples and learn the underlying characteristics and behaviors that distinguish them from benign files. By training on these datasets, machine learning models can develop the ability to identify new and unknown threats (called zero-day threats) based on their similarities to previously seen malware. This enables antivirus software to detect and block previously unseen malware, enhancing overall security.

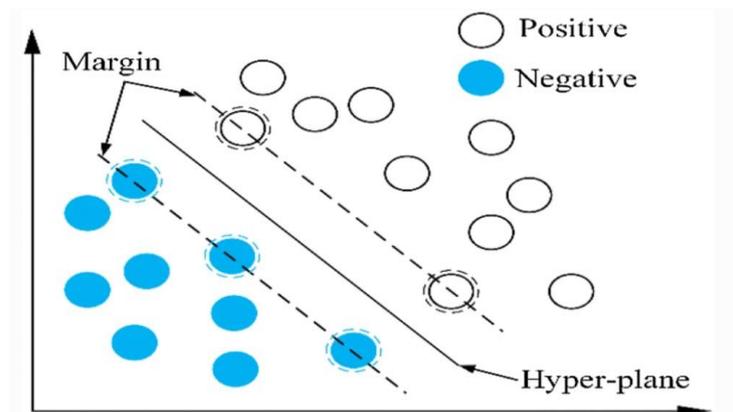
## **.3 Famous ML algorithms used for ransomware detection:**

### **.3.1 Support Vector Machines (SVM)**

SVM is a supervised learning model employed for both classification and regression analysis tasks. The fundamental concept of SVM is to create an optimal decision boundary, referred to as a hyperplane, to segregate the n-dimensional space into distinct classes. This is achieved by mapping input vectors to a higher-dimensional space and constructing a maximal separating hyperplane.

To ensure maximal separation, SVM constructs two parallel hyperplanes on each side of the separating hyperplane, forming a margin. The algorithm then maximizes this margin, striving to maintain the greatest possible distance between the hyperplane and the nearest data points from each class.

SVM is particularly valuable when dealing with high-dimensional data due to its versatile decision function, which can be customized with various Kernel functions. It excels in its ability to generalize, making it suitable for tasks such as ransomware detection, by treating such cases as a classification problem to distinguish between ransomware or benign software[35][36][38]



**Figure 2.1** : Classification of two tapes using SVM [80]

### **.3.2 Decision trees**

A decision tree is a supervised learning model where each internal node corresponds to a feature or attribute, each branch embodies a decision rule, and each leaf node signifies an outcome. The apex of the decision tree is referred to as the root node. The model operates by learning simple decision rules, inferred from the data features, to predict the value of a target variable.

One of the key advantages of decision trees is their simplicity and interpretability, with little data preparation required. They can handle both numerical and categorical data efficiently. However, a significant downside is their sensitivity to small changes in the training data, which can lead to substantial changes in the resulting tree structure. They also have a tendency to overfit or underfit the data, sometimes creating complex trees that do not generalize well. In addressing some of these issues, the Random Forest algorithm introduces an ensemble approach. It creates a 'forest' of decision trees, each trained on a random subset of the data. When a new data point is introduced, each tree in the forest makes a prediction, and the final output is determined by majority vote. This technique improves the overall predictive accuracy and helps to control overfitting, enhancing the stability and robustness of decision trees [34].

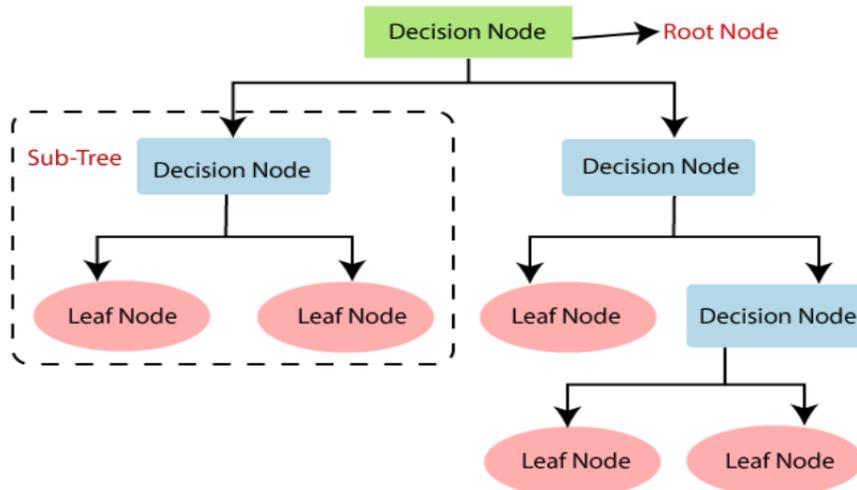


Figure 2.2 : Decision Trees.

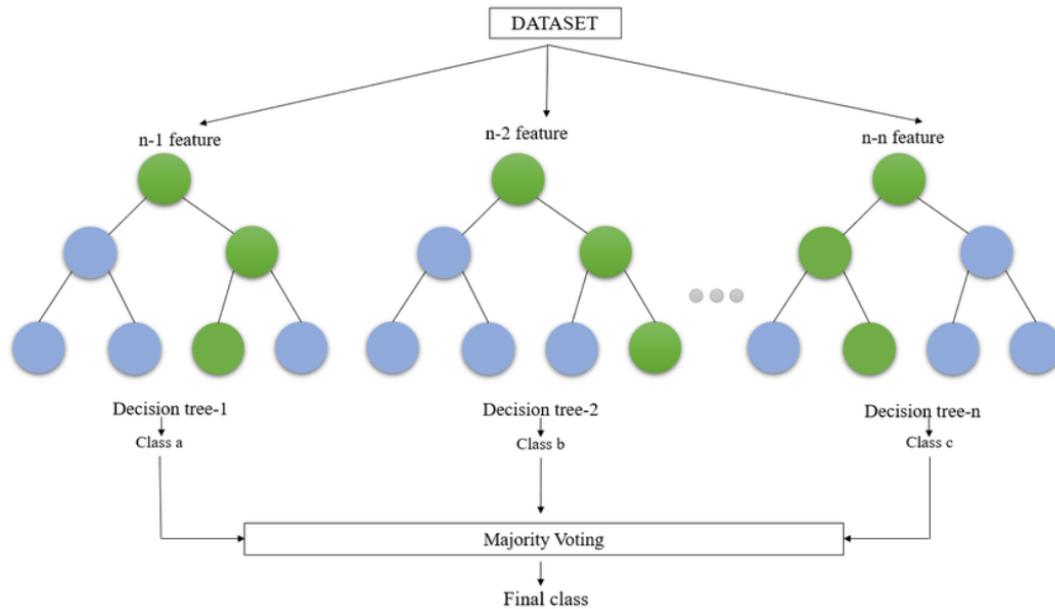
### .3.3 Random Forest

The Random Forest algorithm is an ensemble learning method based on decision trees. It builds numerous decision trees at the training phase, with the final output derived from the mode of classes in the case of classification, or the mean prediction for regression. This algorithm operates on the principles of bootstrapping and feature randomness; it forms subsets from the original dataset and selects a random subset of features at each candidate split in the learning process.

Random Forest is versatile and effective, capable of handling large datasets with high dimensionality and thousands of input variables efficiently. It even has an inherent ability to manage missing values, making it well-suited for real-world datasets that are often incomplete.

This algorithm is particularly adapted at overcoming the overfitting problem common in single decision trees. Overfitting happens when a model learns the training data too well and performs poorly on unseen data. By employing multiple decision trees and averaging their predictions, Random Forest mitigates this risk, ensuring a more robust, reliable, and generalized model.

However, despite its strengths, Random Forest can be computationally expensive and may not offer the same level of interpretability as a single decision tree due to its complexity. Also, the algorithm can be slow in creating predictions once trained, an aspect to consider when dealing with real-time applications. Despite these caveats, Random Forest's robustness and accuracy make it a powerful tool in machine learning [34].



**Figure 2.3 :** Generalized structure for random forest [81].

## .4 Deep learning

Deep learning is a subset of machine learning in artificial intelligence (AI) that mirrors the functioning of the human brain through artificial neural networks. These networks process and interpret complex data, identify patterns, and deliver insights or predictions. The neural networks have self-learning capabilities that improve accuracy and performance over time. Deep learning finds widespread applications, including image and speech recognition, natural language processing, autonomous driving, and cybersecurity.

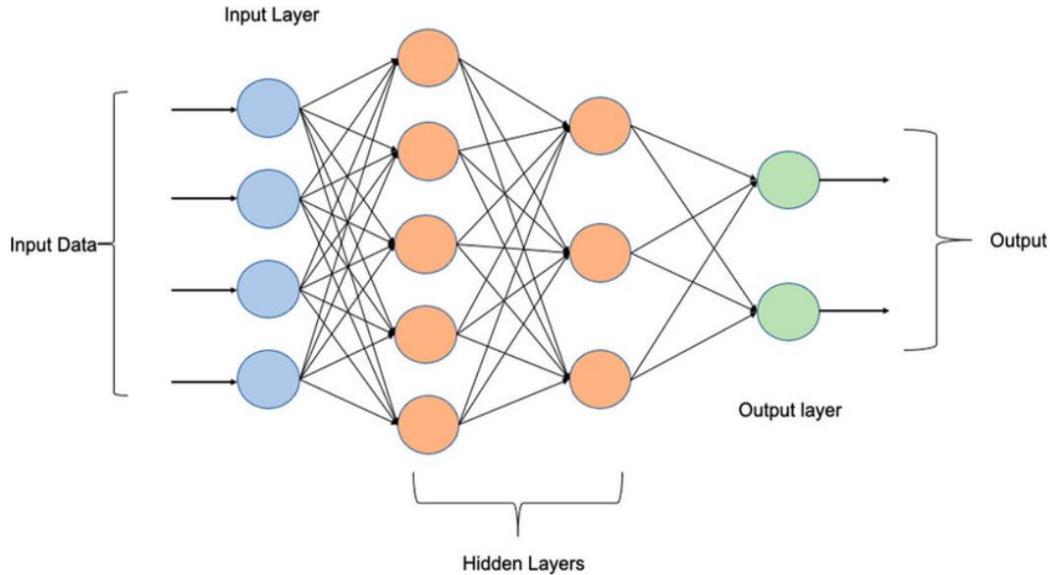
In cybersecurity, it aids in identifying malicious activities, detecting anomalies, and mitigating cyber threats, enhancing the security of digital systems [53][54].

### .4.1 Types of neural networks

#### .4.1.1 FNN

Feed Forward Network also known as Multilayer Perceptron is a type of artificial neural network in which data flows in single direction, moving from the input to the hidden layer (one or many layers) and finally to the output node. FNN is capable of classifying large amounts of data and finding patterns in complex datasets. It has been widely used across multiple domains due to its simplicity and effectiveness in learning complex nonlinear mappings and adapt to new data [46][48]

Within each layer, there are several neurons, and these neurons are connected to all neurons in the preceding and succeeding layers. The connections between neurons have associated weights, which are adjusted during the training process to minimize the error between the network's output and the desired output [48].



**Figure 2.4** : Multi-layer perceptron (MLP-NN) basic Architecture [82]

### **Applications of Feed Forward Neural Networks in cybersecurity**

FNNs have several applications in cybersecurity. One of the most prominent applications is in detecting cyber-attacks is intrusion detection systems (IDS): One of the primary applications of FFNNs in cybersecurity is in the development of intrusion detection systems (IDS). IDS are designed to detect and prevent unauthorized access to computer systems and networks. FFNNs can be used to analyze network traffic and identify patterns that are indicative of an attack. By training an FFNN on a large dataset of network traffic, the network can learn to recognize patterns that are associated with different types of attacks, such as denial-of-service (DoS) attacks, port scanning, and malware infections [49].

They also play a crucial role in malware classification by learning to recognize patterns of API calls associated with malicious behavior. [50]

### **.4.1.2 CNN**

Convolutional Neural Network (CNN) is a type of deep learning models that have proven to be highly effective in areas such as computer vision and classification. CNNs have been successful in various datasets like pictures, audio and text.

CNNs are designed to automatically learn spatial hierarchies of features from data which has a grid-like topology. Examples include image data which can be thought of as a 2-D grid of pixels [42][43][44][45].

#### **How a CNN Works?**

CNN consists of multiple layers which are [44][45]:

#### **Input layer:**

Represents the raw input data which is usually an image which is structured as a grid of pixels.

#### **Convolutional Layers:**

This layer is responsible of learning and extracting features from the input image. This process is performed using filters or kernels that perform convolution operations on the input.

**Pooling layer:**

This layer reduces the spatial size (width and height) of the Input Volume which helps decrease the computational complexity, memory usage and number of parameters. There are various types of pooling like max pooling, average pooling, etc.

**Flattening:**

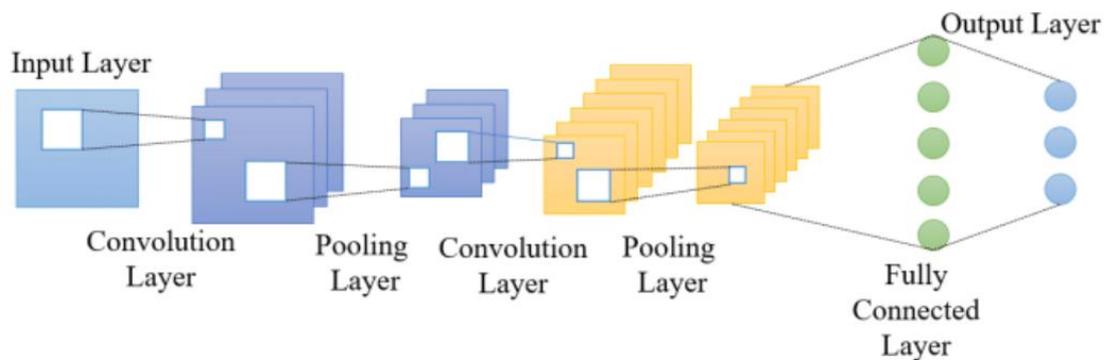
Is a process in a CNN where the output from the last pooling layer is reshaped into a 1-dimensional vector. This transformation allows the subsequent fully connected layers to process the information.

**Fully connected layers (dense layers):**

After several convolutional and pooling layers (See Figure 2.5) and after flattening, fully connected are added to CNN. It serves as the final stages of feature extraction and play a crucial role in capturing global patterns and high-level representations.

**Output layer:**

It is the final layer where the network produces the desired output. The structure and design of the output layer depend on the specific task.



**Figure 2.5 :** Basic architecture of CNN

**4.1.3 RNN**

A recurrent neural network (RNN) is a type of neural network that is designed for processing sequential data such as time series or natural language processing (NLP). Unlike feedforward neural networks which process data in a single pass from input to output, or CNNs which excel on grid-like data, RNNs are designed to work with data where order and temporal dimension matters. RNNs have the capability to utilize their internal state (memory) to process sequences of inputs, which makes them ideal for task such as Sentimental analysis, Speech Recognition and sequence study of the genome and DNA.[44][55]

**How an RNN works?****Input:**

The current input and the previous hidden state are combined to form a new representation. This is typically done by applying a linear transformation (multiplying the input and hidden state by weight matrices) and applying a non-linear activation function, such as the hyperbolic tangent (tanh) or the rectified linear unit (ReLU).

### Hidden state update:

The new representation is used to update the hidden state. The updated hidden state becomes the memory of the network, capturing the relevant information from past inputs. This step is again performed by a linear transformation and an activation function.

### Output computation:

The hidden state is used to compute the output for the current time step. The output can be a prediction, classification, or any other relevant information based on the task at hand.

The process described above is repeated for each time step in the input sequence, allowing the network to capture dependencies and patterns over time. The recurrent connection in the RNN enables information to flow from past time steps to the present, which is crucial for tasks involving sequential data.

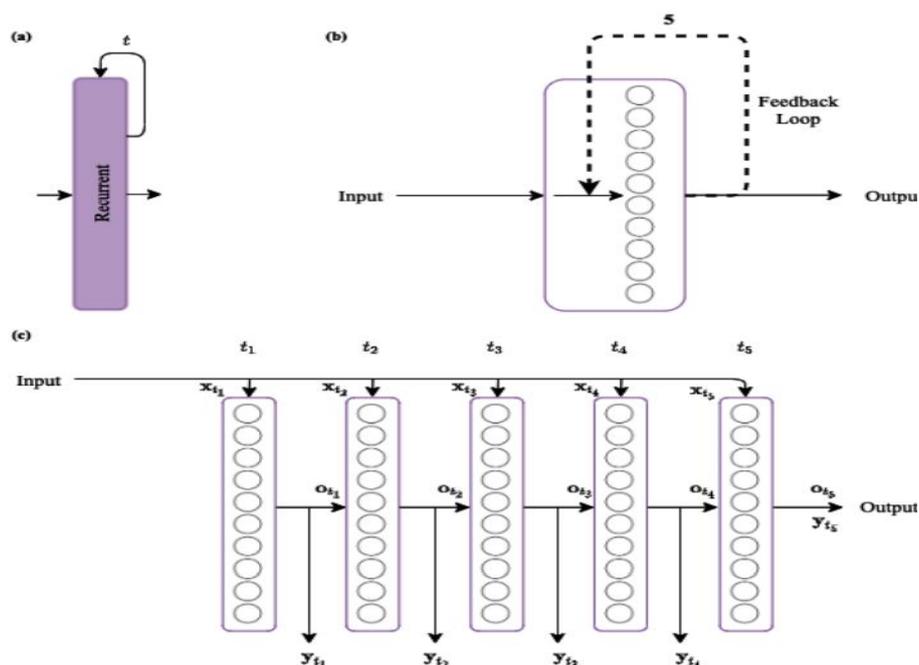


Figure 2.6 : Different representations of recurrent layer [44].

## 4.1.4 LSTM

Long Short-Term Memory (LSTM) neural networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data. This makes them well suited for tasks such as natural language processing, speech recognition, and time series forecasting.

LSTM networks are composed of cells, each of which contains a memory cell and three gates: an input gate, an output gate, and a forget gate. The input gate controls how much new information is stored in the memory cell, the output gate controls how much information is output from the cell, and the forget gate controls how much information is forgotten from the cell.

LSTM networks are able to learn long-term dependencies by using the memory cell to store information over time [57][58].

#### .4.1.5 AutoEncoders

Autoencoders are a type of artificial neural network used for learning efficient codings of input data. They are unsupervised learning models that use the concept of data compression and decompression to learn the identity function, aiming to output a copy of the input. The main idea is to learn a representation (encoding) for a set of data, typically for dimensionality reduction [39].

## .5 Previous studies on ransomware detection

There have been several studies that covered ransomware detection using machine learning, deep learning or combination of both. **Table 2.1** provides a summary of some literature works related to ransomware detection that employ various approaches with diverse datasets.

For instance, in 2019 Bae et al [75] proposed a unique approach for detecting ransomware, distinguishing it from benign files and other types of malware.

The proposed method is based on dynamic analysis of Windows API invocation sequences extracted from the malware samples.

The API sequences are converted into n-gram sequences, which are vectorized using the CF-NCF concept to assign weights to each n-gram based on its frequency in the class and non-class data.

The weighted n-gram vectors are then used as input to six different machine learning algorithms for classification. The experimental results show that the proposed method can effectively detect ransomware among malware and benign files with high accuracy and low false positive rate. The proposed method has several contributions like: providing a new protection mechanism specialized for ransomware detection, using dynamic analysis to extract API sequences and n-gram sequences for classification, introducing the CF-NCF concept to vectorize the n-gram sequences and assign weights to each n-gram, evaluating the performance of six different machine learning algorithms for ransomware detection, achieving high accuracy and low false positive rate in detecting ransomware among malware and benign files.

In 2017, Maniath, S. et al[76] proposed an automated approach to detect ransomware behavior by employing LSTM networks for binary sequence classification of API calls. The authors suggest that the API call sequence of a process can be used as a metric to identify the behavior of a process. They used the common properties of ransomware, such as short-term connection to the Command-and-Control Center, deletion of shadow volumes, and a large number of file system operations, as features to develop a model that can identify ransomware behavior.

The authors suggest that the accuracy of the system can be further improved by modifying the LSTM network structure and using a better dataset with more ransomware samples and benign executables. The proposed approach is expected to improve the automated analysis of a large volume of malware samples.

The authors used a modified sandbox environment to extract API calls from the log and detect ransomware behavior. The proposed method can be extended to general malware classification systems to enhance the automated dynamic analysis of malware samples. The authors suggest that the proposed method can be used in academia and industry to automate malware analysis.

According to Sgandurra et al. [60], EldeRan is a machine learning approach for dynamically analyzing and classifying ransomware. The methods used in the paper include feature selection with the Mutual Information criterion and regularization of the Logistic Regression for preventing overfitting. The machine learning component of EldeRan consists of two phases: feature selection and classification.

The results show that EldeRan achieves an area under the ROC curve of 0.995, which suggests that dynamic analysis can support ransomware detection. The contribution of the paper is the development of a dynamic analysis approach for detecting ransomware, which can help in the early detection of new variants. The paper also outlines some limitations of dynamic analysis for ransomware and proposes possible solutions.

In summary, the paper presents a machine learning approach for dynamically analyzing and classifying ransomware, which can help in the early detection of new variants. The approach uses feature selection and regularization to prevent overfitting. The results show that the approach achieves high accuracy in detecting ransomware.

In 2022, Zahoora et al.[62] proposed a novel framework called "CSPE-R" for detecting zero-day ransomware attacks. The framework consists of five phases: core feature extraction, cost matrix formulation, learning heterogeneous base estimators, estimator selection, and decision aggregation.

The proposed framework uses unsupervised deep Contractive Auto Encoder (CAE) to transform the underlying feature space to a more uniform and core semantic feature space. The CSPE-R ensemble technique explores different semantic spaces at various levels of detail to learn robust features. Heterogeneous base estimators are then trained over these extracted subspaces to find the core relevance between the various families of ransomware attacks. A novel Pareto Ensemble-based estimator selection strategy is implemented to achieve a cost-sensitive compromise between false positives and false negatives. The decision of selected estimators is aggregated to improve the detection against unknown ransomware attacks. The proposed framework performs well against zero-day ransomware attacks and is evaluated using quantitative evaluation metrics such as accuracy, F1 score, FP, TP, and TN. The proposed framework considers host-based features, but in the future, network traffic verification can also be investigated as a feature. The current study focuses on eleven families of ransomware, and an extended version may comprise training the proposed framework with additional ransomware variants.

In 2022, Zahoora et al. [64] proposed a new approach for detecting zero-day ransomware attacks using Zero-shot Learning (ZSL) capabilities. The approach consists of two stages: Attribute Learning (AL) and Inference Stage (IS). In the AL stage, a Deep Contractive Autoencoder (DCAE) is used to extract core features of known and unknown ransomware. The regularization term of CAE helps in penalizing the classifier's sensitivity against the small dissimilarities in the latent space. In the IS stage, a voting-based ensemble classifier is used to find the final prediction. Four combination rules are utilized to find the final prediction. The proposed approach shows reasonable performance against zero-day attacks compared to conventional machine learning techniques. The approach has demonstrated significant improvement in detecting zero-day attacks (recall = 0.95) and reducing False Negative (FN = 6). The paper also discusses the limitations of the proposed approach, such as the homogeneity of the data used in the experiments and the lack of family classification of ransomware. The paper concludes that the proposed approach can effectively detect zero-day ransomware attacks and can be used as a potential solution for enhancing the performance of network intrusion detection systems.

Works	Dataset used	Analysis	Feature used	Algorithm / Architecture	Accuracy
(Bae et al., 2019) [75]	1000 Ransomware files 300 Malware files 300 Benign files	Dynamic	API SEQUENCE	<b>RF</b>	<b>98.65%</b>
				<b>LR</b>	<b>92.9%</b>
				<b>NB</b>	<b>93.53%</b>
				<b>SVM</b>	<b>85.94%</b>
				<b>KNN</b>	<b>96.22%</b>
				<b>SGD</b>	<b>97.64%</b>
(Maniath, S. et al,2017)[76]	157 ransomware samples and benign executables 15 families of ransomware	Dynamic	API CALL SEQUENCES	<b>LSTM</b>	<b>96.67%</b>
(D. Sgandurra. et al, 2016)[60]	It contains: 1524 samples divided into 582 Ransomware and 942 Benign 11 families of ransomware	Static and dynamic	API CALL DROP REG FILES FILES_EXT DIR STR	<b>Logistic Regression</b>	<b>94%</b>
(Zahoora, U et al,2022) [62]	It contains of 582 ransomware and 942 goodware instances. Ransomware samples are further classified into 11 families	Static and dynamic	API CALL DROP REG FILES FILES_EXT DIR STR	<b>unsupervised deep Contractive Auto Encoder</b>	<b>93.28%</b>
Zahoora, U et al,2022) [64]	It contains of 582 ransomware and 942 goodware instances. Ransomware samples are further classified into 11 families	Static and dynamic	API CALL DROP REG FILES FILES_EXT DIR STR	<b>Deep Contractive Autoencoder</b>	<b>95%</b>

**Table 2.1** : Previous studies comparison

## **.6 Conclusion**

In conclusion, this chapter highlights the significance of machine learning algorithms and deep learning models for ransomware attack detection. It discussed the applications and benefits of these techniques, presenting notable algorithms like Support Vector Machines, Decision Trees, and Random Forest, as well as various neural networks such as FNN, CNN, RNN, LSTM and Autoencoders.

Previous studies have demonstrated promising results, achieving high accuracy rates in ransomware detection. However, there is still a notable issue of high false positives, where legitimate software is mistakenly detected as ransomware. Nevertheless, the implementation of neural networks demonstrates promising capabilities in detecting and mitigating cyber threats, enabling the identification and response to ransomware and the detection of zero-day threats. By surpassing the limitations of rule-based methods, neural networks facilitate the development of more robust detection systems.

## Chapter 3 Proposed approach

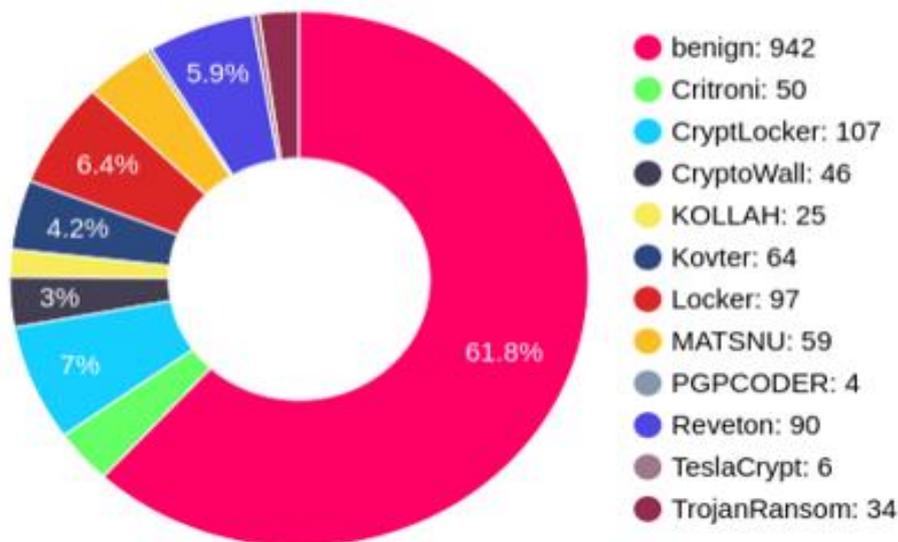
### .1 Dataset Collection and Preprocessing

#### .1.1 Description of the dataset

The dataset used in this study is “ransomwaredataset2016” [74] developed by Sgandurra et al[60] as training dataset to construct the EldeRan model. The dataset is obtained by analyzing the samples using cuckoo sandbox. Malicious samples are labeled as ransomware (one or positive class), and benign samples are labeled as goodware (zero or negative class). The dataset contains 582 ransomwares belonging to 11 families (See **Figure 3.1**) and 942 goodware [60][62].

The assembled samples are the most widespread variants of the ransomware downloaded from VirusShare database, and the mainstream is Crypto Ransomware type. The goodware samples are prepared from reliable sources. Goodware samples include generic utilities for Windows (e.g., zipper, password managers, etc.), drivers, browsers (the most popular ones), file utilities (DropBox, file search, etc.) multimedia tools (music, video, etc.), developer’s tools (Eclipse, notepad++, etc.), games, network utilities, paint tools, databases, emulator and virtual machines monitors, office tools, etc [60].

Separately these applications are executed for thirty seconds in a sandbox setting. Sgandurra et al. considered only host-based features while structuring the samples (See **Table 3.1**)



**Figure 3.1** : Number of samples for each family of Ransomware

Class of Features	Details	Number of features
API calls	the traces of invocations of native functions and Windows API calls	232
Reg Operations	Registry Key Operations (read, open, write, and delete operations)	6622
File Operations	File System Operations (read, open, write, and delete operations)	4141
File Extension Operation	the set of file operations performed per File Extension	935
Dir Operations	the set of operations performed on directories, in particular the enumeration and creation	2424
Dropped Files	the set of files that are dropped by an application during installation	346
Strings	the strings embedded in the binary	16267

**Table 3.1** : Dataset features classification.

All class of features are dynamic features except **Strings**.

## **.1.2 Preprocessing steps and feature extraction techniques applied to the data:**

the first step in our work is dividing the data into training and testing set. We split the ransomware families into seen and unseen classes to build a robust zero-day ransomware detection model (See **Table 3.2**). The seen families are used for training and validation purposes, and the unseen classes are reserved as zero-day attacks for testing purposes.

We used same splitting strategies used in Zahoora et al. [62] to evaluate the performance of our model with their results.

Seen families	Unseen families
CryptoLocker	Trojan Ransom
CryptoWall	Tesla Crypt
Critroni	PGPCODER
KOLLAH	Reveton
MATSNU	
Kovter	
Locker	

**Table 3.2** : Train test split.

	Training set	Testing set
<b>Ransomware samples</b>	448	134
<b>Goodware samples</b>	808	134

**Table 3.3** : Number of samples of training and testing sets.

### **.1.3 Feature selection:**

For each class of features, we select top features using Univariate feature selection, a common method used in machine learning to select the most relevant features from a given dataset. It evaluates each feature independently and selects the top features based on certain statistical measures. The goal is to choose the features that have the strongest relationship with the target variable or the most discriminatory power.

There are several statistical measures commonly used in univariate feature selection, we applied the following three statistical measures for the feature selection phase:

#### **.1.3.1 ANOVA F\_Value:**

This method is applicable when the target variable is categorical. It calculates the F-value statistic for each feature by comparing the means of the feature values across different categories of the target variable.

#### **.1.3.2 Chi\_Squared\_Test:**

This method is also used when the target variable is categorical. It measures the dependence between each feature and the target variable by computing the chi-squared statistic.

#### **.1.3.3 Mutual Information:**

Mutual information measures the amount of information shared between a feature and the target variable. It computes the mutual information score for each feature and the target variable, which is used to rank the features and select the top k features with the highest scores.

The resulted set of features for each method are evaluated using a Logistic Regression model to select the best set for the training phase.

<b>Class of features</b>	<b>Number of features selected</b>	<b>Percentage of features selected</b>
<b>Api calls</b>	200	86.21%
<b>Dropped files</b>	200	57.80%
<b>Reg operations</b>	100	1.51%
<b>File operations</b>	200	4.83%
<b>File extension operations</b>	100	10.70%
<b>Dir operations</b>	100	4.13%
<b>General</b>	400	2.72%
<b>Strings</b>	0	0%

**Table 3.4** : Selected features per class.

## .2 Model Design

### .2.1 Ensemble learning:

Ensemble learning is a machine learning technique that combines multiple models (also known as base learners) to make better predictions. The idea behind ensemble learning is that the combined predictions of multiple models can be more accurate and robust than the predictions of individual models.

There are different types of ensembles learning methods (See **Figure 3.2**):

#### .2.1.1 Bagging (Bootstrap Aggregating):

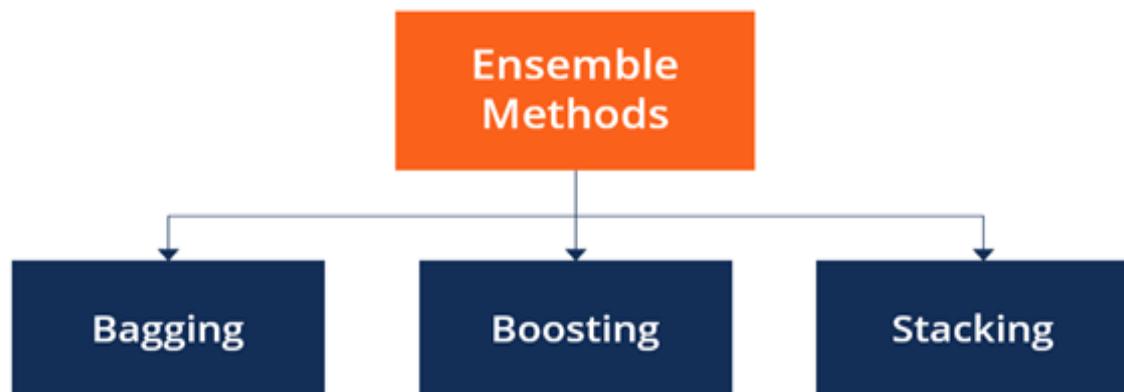
In bagging, multiple models are trained independently on different subsets of the training data. Each model makes a prediction and the final prediction is obtained by aggregating the predictions of all base learners, such as by majority voting (for classification tasks) or averaging (for regression tasks). Random Forest is an example of a bagging ensemble algorithm.

#### .2.1.2 Boosting:

In boosting, models are trained in sequence manner, with each model trying to correct the errors of the previous model. Boosting assigns higher weights to misclassified instances, allowing subsequent models to focus more on the difficult instances. The final prediction is the weighted sum of the predictions of all the models.

#### .2.1.3 Stacking:

In stacking, multiple models are trained and their outputs are used as features for a meta-model, which makes the final prediction.



**Figure 3.2 :** Different types of ensembles learning methods

## 2.2 Proposed Method:

The proposed approach incorporates bagging with feature selection to capture different aspects of the data and improve model performance. This combination exposes each base learner to a different subset of the training data while reducing model complexity. During the aggregation phase, a stacking technique is employed, where a new dataset is constructed using the predictions of the seven base learners as features. This dataset is then used to train the final meta model, which leverages the collective knowledge of the base learners to make robust predictions. (See Figure 3.3).

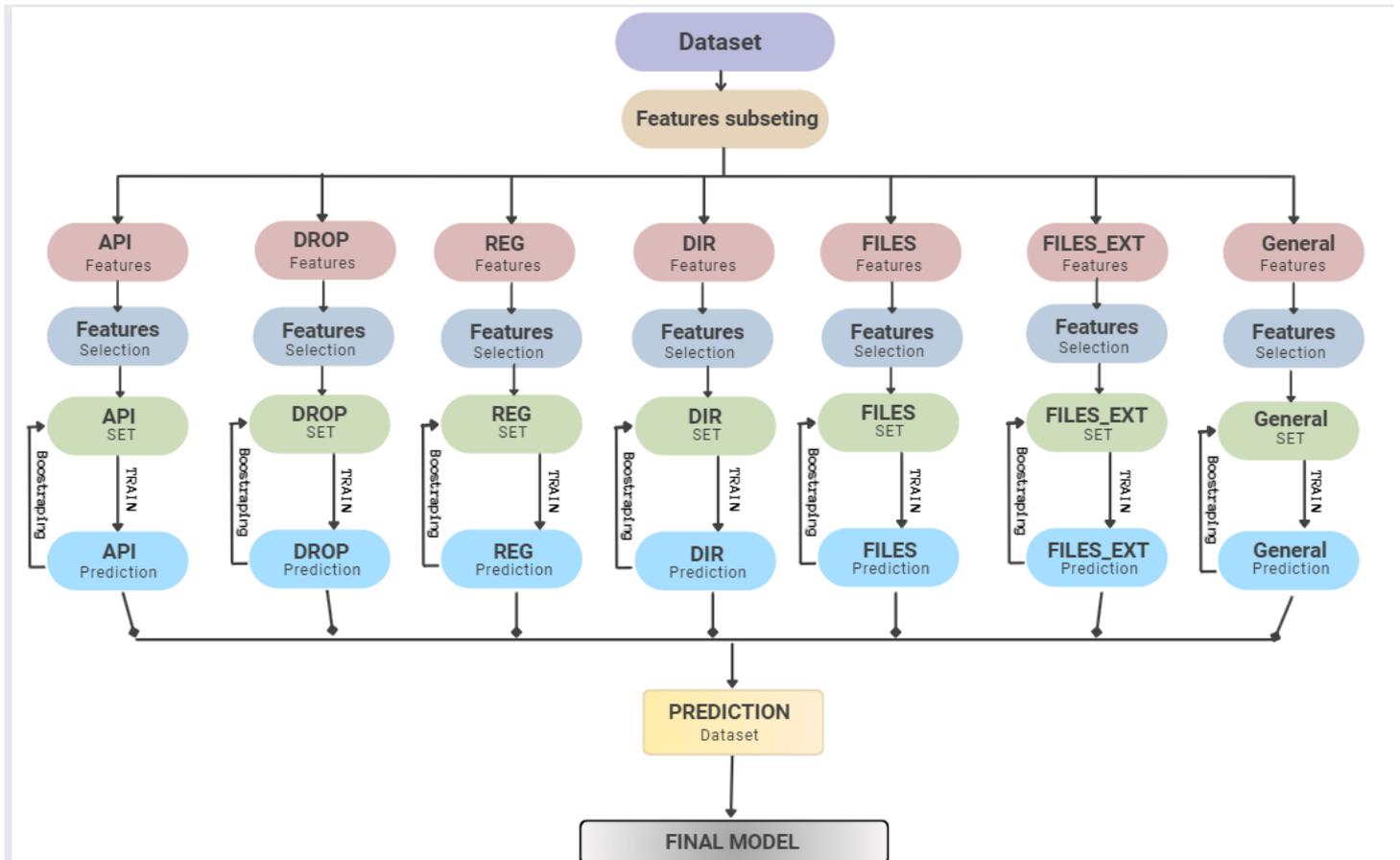


Figure 3.3 : Proposed method

For the base learners, feed-forward neural networks are utilized. This architecture is well-suited for handling high-dimensional data and capturing complex nonlinear relationships between input features and target labels. The base models consist of fully-connected layers (Dense layers), allowing for the learning of intricate patterns and representations within the data.

Batch normalization layers are incorporated into the neural network architectures to standardize the input data. This technique accelerates convergence by reducing internal covariance shift and improves model accuracy by normalizing inputs to each neuron, thus stabilizing weight updates during training.

Additionally, Dropout layers are included in the base models as a regularization technique to prevent overfitting. By randomly dropping out a certain proportion of input units during each training iteration, the models are encouraged to learn more robust and generalizable features.

A total of seven base learners are trained in this approach. Six of these base learners are trained on different classes of features, such as API calls, file extensions operations, and directories operations. Each of these base learners specializes in modeling a specific class of features. The seventh base learner is trained on the integration of these six classes of features, allowing it to infer relationships between different classes of features. All of these base learners, including the final meta model, are feed-forward neural networks.

By utilizing feed-forward neural networks with different base learners, feature selection, and regularization techniques, the proposed approach aims to enhance the understanding of ransomware behaviors and improve the overall accuracy of ransomware detection.

### .3 Model Training and Validation

All the training was conducted exclusively in the Colab environment without any additional accelerators. The base estimators were developed using the Keras library, while the Random Forest and Logistic Regression models were implemented using the Scikit-learn library. Additionally, feature selection was performed using the SelectKBest module from the Scikit-learn library.

The evaluation of the developed model involves the utilization of the following evaluation metrics:

#### 1. Accuracy:

Measures the number of correct predictions made by a model in relation to the total number of predictions made, calculated as

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

#### 2. Recall:

Also known as sensitivity or true positive rate, it measures the framework's ability to correctly identify positive instances. It is defined as

$$Recall = \frac{TP}{TP + FN}$$

#### 3. Precision:

It indicates the precision of positive predictions made by the framework and is computed as  $TP / (TP + FP)$

$$Precision = \frac{TP}{TP + FP}$$

#### 4. F1-Score:

The F1 score is a single value that combines precision and recall, providing a balanced evaluation of a classification models performance. It is calculated using the following formula:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

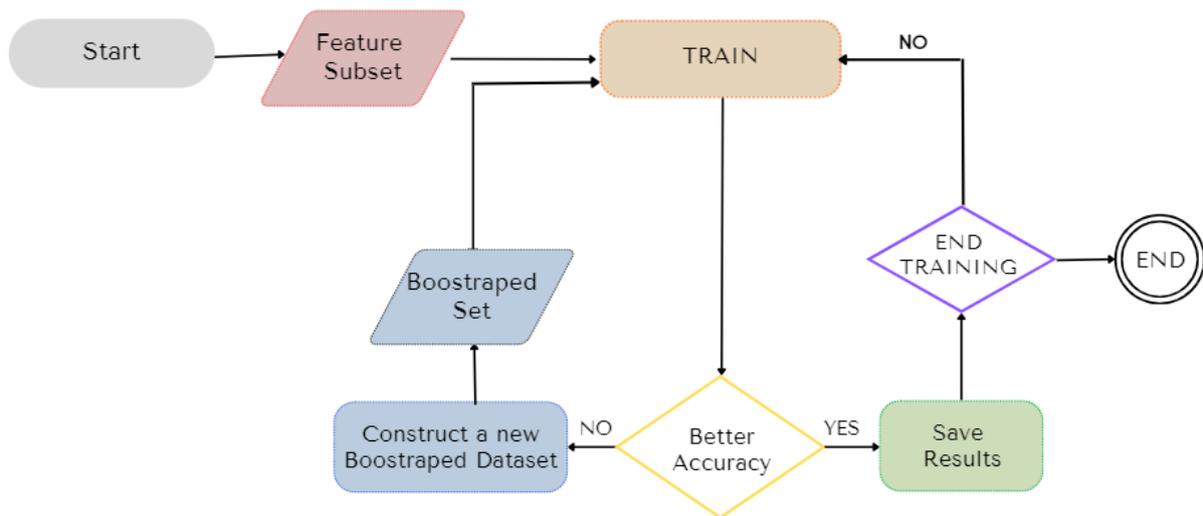
Where:

**TP:  
True Positive**                      The number of positive instances that were correctly predicted as positive by the model.

**TN:  
True Negative**                      The number of negative instances that were correctly predicted as negative by the model.

**FP:  
False Positive**                      The number of negative instances that were incorrectly predicted as positive by the model.

**FN:  
False Negative**                      The number of positive instances that were incorrectly predicted as negative by the model.



**Figure 3.4 :** Flowchart of base learners training process

As shown in **Figure 3.4**, the training process involves training each base learner using the bootstrap sampling method to achieve improved results through multiple iterations, we employed a loop-based approach to train the models using different bootstrapped datasets.

We repeatedly generate new bootstrap samples and train the models on these varied subsets. Each iteration involved assessing the performance of the models and comparing the results obtained with the previous iterations.

## .4 Results:

The objective of our project is to develop a robust model capable of accurately detecting ransomware without affecting the normal utilization of a PC.

MODEL	TN	FP	FN	TP	ACC(%)	Recall(%)	Precision(%)
Api model	134	0	13	121	95.15	90.3	100
Drop model	108	26	19	115	83.21	85.82	81.56
Reg model	106	28	6	128	87.31	95.52	82.05
File model	110	24	22	112	82.84	83.58	82.35
File_ext model	111	23	14	120	86.19	89.55	83.92
Dir model	104	30	4	130	87.31	97.01	81.25
General model	133	1	14	120	94.40	89.55	99.17
Final model	132	2	5	129	97.39	96.27	98.47

**Table 3.5** : Training result of base learners and the final prediction model.

In our evaluation, we employed some widely acknowledged ML algorithms as the benchmark to assess the quality and accuracy of our results.

In addition, we conducted a comparative analysis by referring to previous studies in the field that used the same dataset.

MODEL	TN	FP	FN	TP	ACC (%)	Recall (%)	Precision (%)	F1 Score (%)
Proposed method	132	2	5	129	97.39	96.27	98.47	97.36
RF1	131	3	15	119	93.28	88.81	97.54	92.97
RF2	131	3	10	124	95.15	92.54	97.64	95.02
SVM	133	1	24	110	90.67	82.09	99.10	89.80
LR	134	0	9	125	96.64	93.28	100	96.52
Zahoor et al[62]	117	17	1	133	93.28	99.25	88.67	93.66
Zahoor et al[64]	120	13	6	12	95	92.8	90.8	91.8
Khan et al[65]	141	9	27	123	88	82	93.18	87.23

**Table 3.6** : Comparison with various ML algorithms and current techniques in the field.

As shown in **Table 3.6**, our model demonstrates an overall superior performance compared to most techniques. Our model exhibits a remarkable capability detecting both ransomware and goodware, ensuring that PCs can be used normally without falsely identifying benign software as ransomware and a lower chance of getting infected with ransomware.

**Note:**

RF1(criterion='gini', max\_features = 'sqrt', n\_estimators=10000, random\_state = 0).

RF2(criterion='gini', max\_features = 1.0, n\_estimators=10000, random\_state = 0).

The following table provides a comprehensive overview of the performance achieved by our developed model for detecting different ransomware families when trained solely on data from the remaining ransomware families, along with a comparison to the results of some widely used ML algorithms.

family	samples	RF1	RF2	SVM	LR	EldeRAN(400 ft) [62]	EldeRAN(100 ft) [62]	Proposed method
Critoni	50	90	92	90	92	92	98	100
CryptoLocker	107	91.59	87.85	91.59	91.58	90.65	96.26	97.19
CryptoWall	46	71.74	76.09	71.74	73.91	73.91	91.30	100
KOLLAH	25	76	76	72	76	76.00	96	92
Kovter	64	90.63	90.63	79.69	90.63	89.06	89.06	95.31
Locker	97	87.62	91.75	81.44	86.60	85.57	91.75	90.72
Matsnu	59	89.83	93.22	86.44	93.22	91.53	98.31	93.22
Pgcodcr	4	100	100	100	100	100	75	100
Reveton	90	90	95.56	73.33	95.56	91.11	88.89	95.56
TeslaCrypt	6	83.33	83.33	83.33	83.33	83.33	100	83.33
Trojan_Ransom	34	88.24	85.29	88.24	88.24	76.47	94.12	88.24
<b>Weighted_AVG</b>	<b>582</b>	<b>87.80</b>	<b>89.35</b>	<b>82.47</b>	<b>89.18</b>	<b>87.11</b>	<b>93.30</b>	<b>94.43</b>

Table 3.7 : Cross validation results compared with various ML algorithms

**Sandboxing:**

Sandboxing is a security measure that establishes a controlled environment for executing potentially untrusted or malicious code by isolating an application or process from the rest of the system. It serves as a virtualized environment in which code can operate securely without impacting the underlying system or other applications [61].

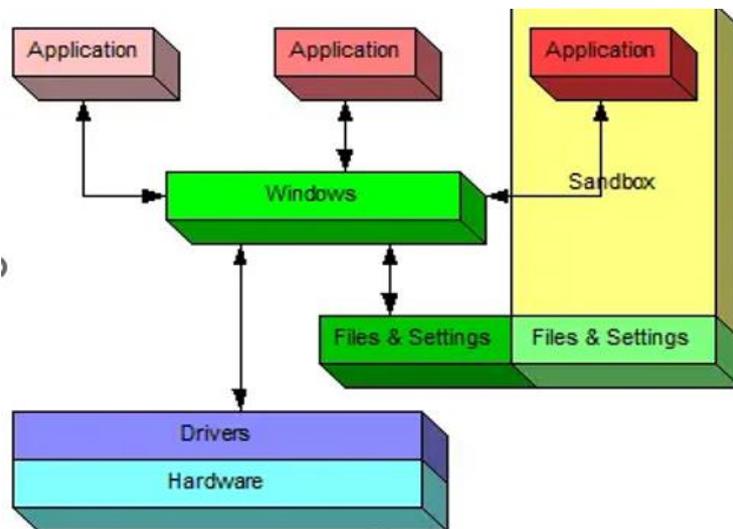


Figure 3.9 : Windows sandbox\_pic from askleo.com

Cuckoo Sandbox is an automated system for analyzing malware that employs sandboxing techniques and is available as open-source software. The tool offers a regulated setting to run potentially harmful files or URLs, enabling security experts and researchers to scrutinize and comprehend their actions.

The Cuckoo Sandbox software application is designed to monitor the behavior of analyzed files, capture network traffic, and observe system-level interactions in order to identify and analyze potential security threats. The tool offers significant insights into malware behavior, encompassing details on file modifications, network communication, registry alterations, and other indicators of compromise.



Figure 3.10 : Cuckoo sandbox logo

Security professionals can safely study and analyze potentially malicious software by utilizing sandboxing techniques such as Cuckoo Sandbox. This approach allows for a controlled environment that mitigates the risk of compromising the security of their systems.

### .5 Implementation

A trained Deep Learning model for Ransomware detection can be seamlessly integrated into an antivirus software product or deployed as a SaaS solution exposing a REST API. In this section we present an implementation for real world use of the model.

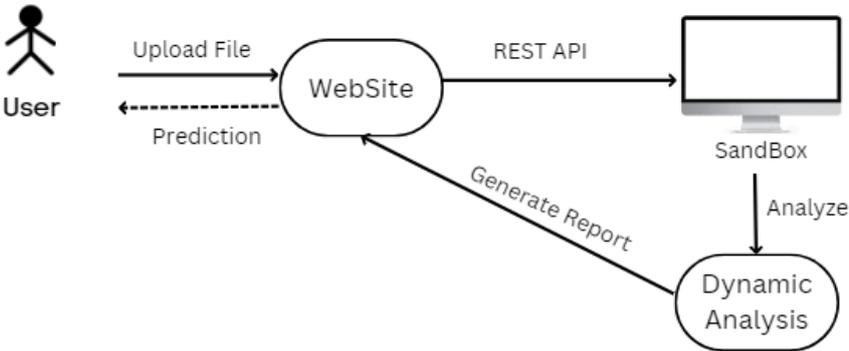
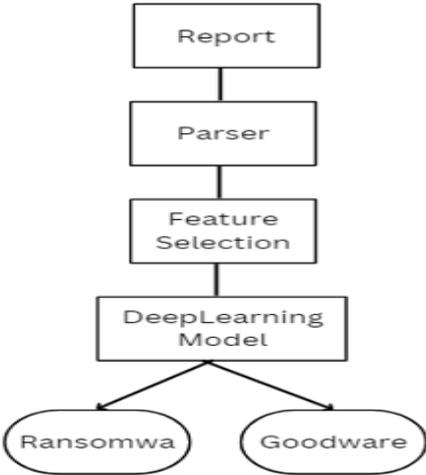


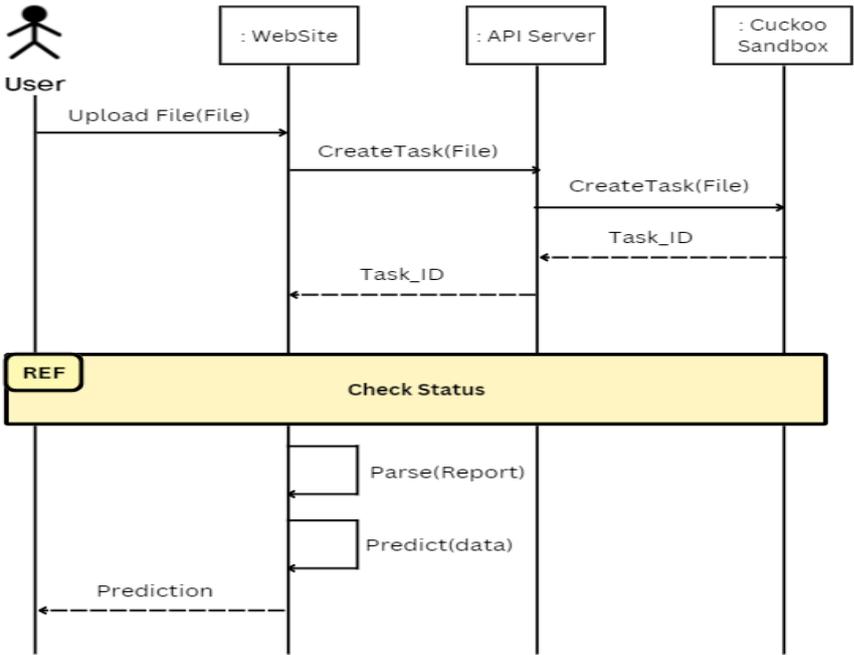
Figure 3.5 : User File Prediction Workflow: From Upload to Final Decision.

In Figure 3.5 , We used Cuckoo Sandbox to monitor a virtual machine running Windows XP. Suspicious files submitted by users were sent for analysis through the Cuckoo REST API. The resulting analysis report is then transferred back to the application for processing. Based on this report, the application provided predictions to the users regarding the nature of the file (e.g., ransomware or goodware). Figures 3.7 and Figure 3.8 explain the process in details.



**Figure 3.6 :** Prediction mechanism using the report

**Figure 3.6** provides an overview of the website's backend workflow, showcasing the key steps of report parsing, feature selection, and final prediction. To ensure compatibility with the dataset used, we have developed a custom parser that generates a data vector with identical structure and features. The analysis report is parsed using this parser, enabling the extraction of relevant information. Subsequently, feature selection is applied using stored indexes. Finally, the integrated model utilizes the selected features to generate the final prediction. This well-organized process guarantees efficient and accurate file classification.



**Figure 3.7 :** Global Sequence diagram.

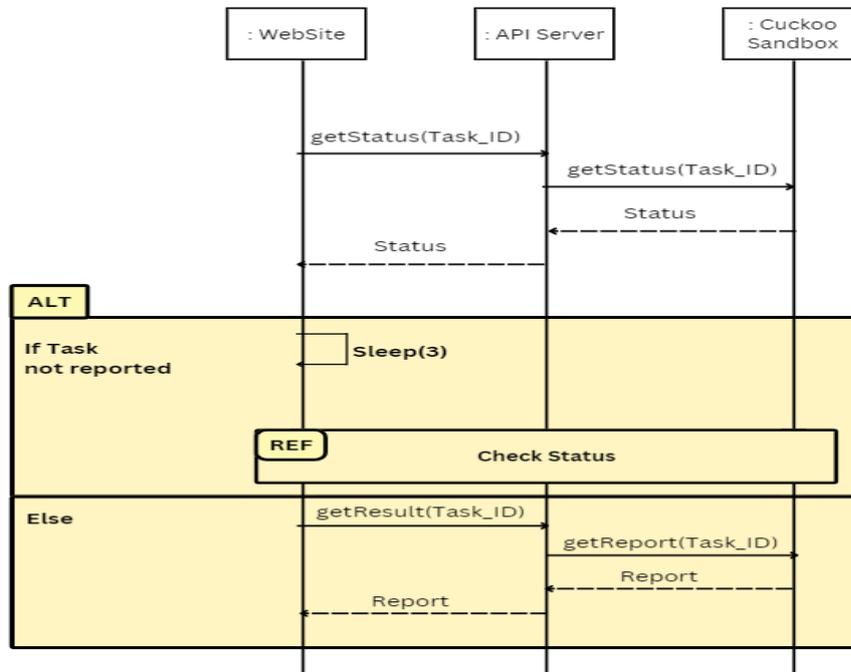


Figure 3.8 : Check Status Sequence Diagram

**Ransomware Detection Results:**

This table presents the findings of a ransomware detection analysis conducted using the trained model. It encompasses a combination of samples from the dataset and newly encountered samples that were not previously observed.

Sample	Included in the dataset?	Detected
Critoni	Yes	YES
CryptoLocker	Yes	YES
Kovter	Yes	YES
Locker	No	YES
Petya	No	YES
PolyRansom	No	YES
RagnarLocker	No	No
Locky	Yes	YES
Matsnu	Yes	YES
Radamant	No	YES
Satana	No	YES
Cryptowall	Yes	YES

XData	No	YES
Revil	No	YES
Ryuk	No	YES
TeslaCrypt	Yes	YES
ViraLock	No	No
TrojanRansom	Yes	YES
WannaCry	No	YES
Xyeta	No	YES
BadRabbit	No	No
Cerber	No	No
RedBoot	No	No
Blacksod	No	No

**Table 3.8** : Analysis of Ransomware Samples and Detection Status.

The results provide insights into the model's generalization capabilities and its effectiveness in identifying previously unseen and newer ransomware variants.

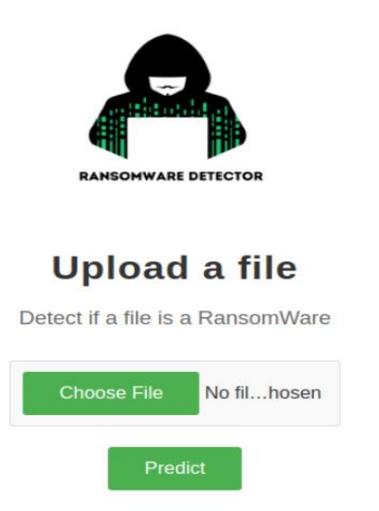
#### **Goodware Detection Results:**

<b>Sample</b>	<b>Type</b>	<b>Detection</b>
7zip	File Compression	Goodware
WinRaR	File Compression	Goodware
CCleaner	System Utility	Goodware
Adobe Reader	PDF Reader	Goodware
Firefox	Web Browser	Goodware
Foxit Reader	PDF Reader	Goodware
Media Player HC	Media Player	Goodware
Notepad ++	Text Editor	Goodware
Opera	Web Browser	Goodware
VLC	Media Player	Goodware

**Table 3.9** : Analysis of Software Samples and Detection Results.

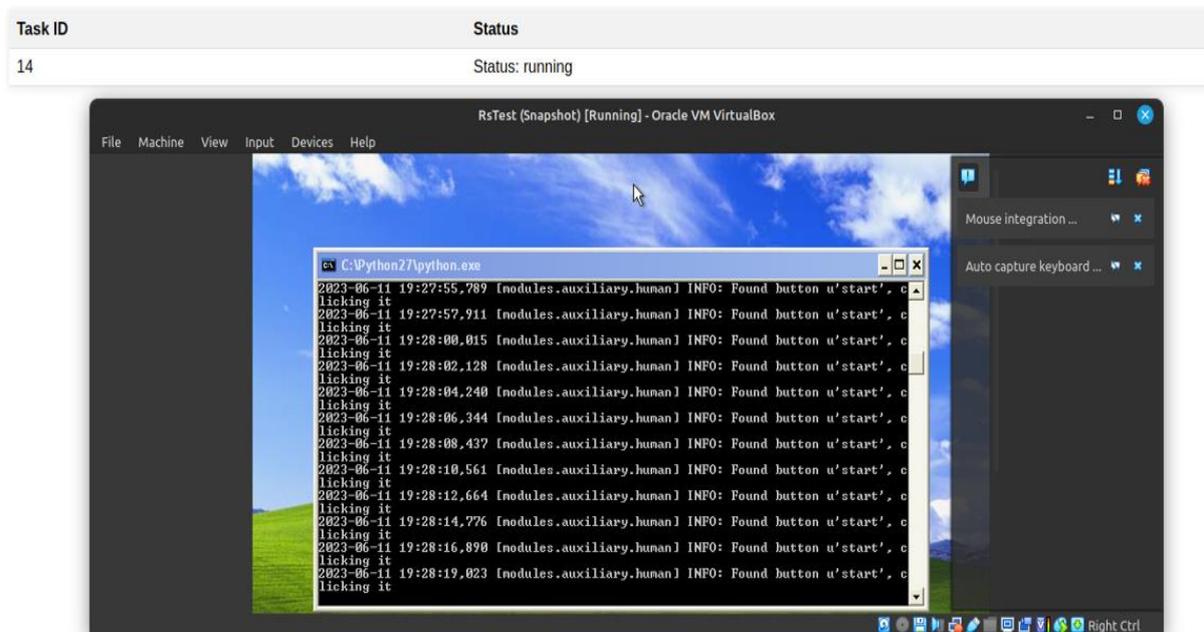
In this analysis, all the listed software applications, including 7zip, WinRAR, CCleaner, Adobe Reader, Firefox, Foxit Reader, Media Player HC, Notepad++, Opera and VLC, were correctly classified as Goodware. This indicates that the model accurately recognizes safe and legitimate software, enabling users to use their computers normally.

### Application Screenshots:



**Figure 3.11** : Upload a File for analysis

The file upload page (See **Figure 3.11**) provides a user-friendly interface for users to submit suspicious files for analysis and prediction. This feature allows users to determine if file is potentially a ransomware or not, using the training model.



**Figure 3.12** : Software execution in the VM

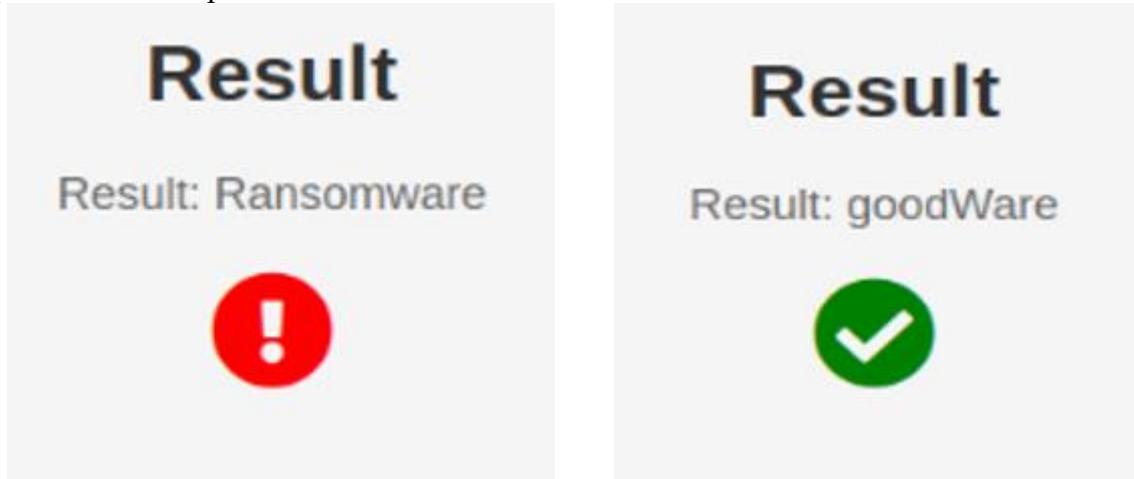
This screenshot captures the execution process of an uploaded file within a secure virtual machine (VM) environment. The status values are updated in real-time as the analysis progresses.

**Pending:** indicating the file is in the queue

**running:** when the execution process is active

**completed:** finished the analysis within the VM

**reported:** This status indicates that the execution data and analysis results have been fully processed and reported



**Figure 3.13** : Successful Classification: Ransomware (Tslacypt) and Goodware (7zip)

The combined prediction screenshot (See Figure 3.13) demonstrates the model's precise classification of the uploaded files, labeling Tslacypt as ransomware and 7zip as goodware. This accurate prediction emphasizes the model's advanced capabilities in distinguishing between ransomware threats and legitimate files, affirming its reliability and reinforcing its suitability for real-world applications.

## .6 CONCLUSION:

In conclusion, the evaluation of our developed model for ransomware detection has yielded promising results, outperforming established methods such as Random Forest (RF), Logistic Regression (LG), and Support Vector Machines (SVM). The model demonstrated superior performance across multiple performance metrics, including accuracy, precision, recall. This indicates its effectiveness in accurately identifying and distinguishing ransomware instances. The robustness and superior performance of our model highlight its potential as a reliable solution for ransomware detection. These findings contribute to the advancement of the field and pave the way for more effective cybersecurity measures against ransomware threat.

## *Conclusion*

Because of the potential profits, ransomware has attracted the attention of numerous cyber-criminals, leading to its rapid evolution and the development of sophisticated samples. To make matters worse, these sophisticated ransomware variations are built to avoid being caught by traditional detection methods such as signature-based used in AV software's.

To effectively protect against ransomware attacks and stay up with the ever-changing threat landscape, it is crucial to invest in powerful ransomware detection approaches that integrate behavioral analysis, machine learning, and heuristic algorithms.

By leveraging the power of behavioral analysis and deep learning, the goal of this research is to develop a robust model for detecting ransomware by training multiple specialized models on various aspects of the data, including API calls, file operations, and registry key operations, all in an effort to minimize false positives while keeping high detection rate with state-of-the-art statistical methods. We used stacking to effectively recognize the class of the samples while taking into account diverse aspects and behaviors, and we trained the models using bootstrapping approaches to offer different perspectives on the underlying patterns and relationships within the data.

The study's findings indicate the effectiveness of the developed model in identifying ransomware samples, the model demonstrated a high level of accuracy, precision and recall in detecting Zero-day Ransomware attacks.

In addition to the development of our ransomware detection model, we conducted comparative evaluations with widely used machine learning algorithms, including logistic regression, support vector machines (SVM), and random forest along with some state of the art approaches.

The results of our comparative analysis revealed that our model outperformed the other algorithms in most of the evaluated aspects. Our model outperformed logistic regression, SVM, and random forest in terms of accuracy and recall indicating an excellent detection capabilities and exhibited better overall performance in identifying ransomware attacks, furthermore it showed a superior precision to most established approaches insuring that legitimate software's are not mistakenly flagged as malicious.

This study's results demonstrate the usefulness of a model that integrates behavioral analysis, deep learning, and bootstrapping techniques, and thus contribute to the development of improved ransomware detection methods.

However, it is important to acknowledge the limitations of the study. One limitation is the inability to analyze and detect silent or dormant ransomware samples that may remain inactive for a certain period or require user interaction before revealing their malicious behavior. In such cases, the analysis fails to extract their features. Additionally, the model currently does not utilize static features, which could be incorporated to improve the detection rate. Moreover, the absence of network features in the dataset limits the ability to detect newer ransomware behaviors, such as data exfiltration, communication with C&C server, or infections within the network. Therefore, incorporating network-related indicators would be valuable for a more comprehensive detection approach. Furthermore, the small size of the

dataset results in incomplete coverage of all ransomware behaviors. Expanding the dataset to include a wider range of samples would contribute to the continuous improvement and advancement of ransomware detection methodologies. A hybrid approach that combines both behavioral and static analysis techniques could potentially provide a more comprehensive and accurate detection mechanism for ransomware. Considering these limitations, future research should focus on addressing these areas to further improve ransomware detection methodologies.

## References

- [1] Drake, V. (2022, July 29). *The history and evolution of ransomware attacks*. Flashpoint. <https://flashpoint.io/blog/the-history-and-evolution-of-ransomware-attacks/>
- [2] Harford, I. (n.d.). *4 types of ransomware: Examples of past and current attacks*. TechTarget. <https://www.techtarget.com/searchsecurity/feature/4-types-of-ransomware-and-a-timeline-of-attack-examples>
- [3] Cimpanu. (2019, April 9). *Reveton ransomware distributor sentenced to six years in prison in the UK*. ZDNET. <https://www.zdnet.com/article/reveton-ransomware-distributor-sentenced-to-six-years-in-prison-in-the-uk/>
- [4] Kyurkchiev, N., Iliev, A., Rahnev, A., & Terzieva, T. (2019). A NEW ANALYSIS OF CRYPTOLOCKER RANSOMWARE AND WELCHIA WORM PROPAGATION BEHAVIOR. SOME APPLICATIONS. III. *Communications in Applied Analysis*, 23(2), 359–382. <https://doi.org/10.12732/caa.v23i2.7>
- [5] Buckbee, M. (n.d.). *Cryptolocker: Everything you need to know*. Varonis. <https://www.varonis.com/blog/cryptolocker>
- [6] *Cryptowall ransomware*. KnowBe4. (n.d.-a). <https://www.knowbe4.com/cryptowall>
- [7] Adamov, A., & Carlsson, A. (2017). *The state of ransomware. Trends and mitigation techniques*. <https://doi.org/10.1109/ewdts.2017.8110056>
- [8] Oz, H., Aris, A., Levi, A., & Uluagac, A. S. (2022). A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions. *ACM Computing Surveys*, 54(11s), 1–37. <https://doi.org/10.1145/3514229>
- [9] Popoola, Segun, Iyekekpolo, Ujioghosa, Ojewande, Samuel, Sweetwilliams, Faith, John, Samuel, & Atayero, Aderemi. (2017). *Ransomware: Current Trend, Challenges, and Research Directions*.
- [10] Keijzer, N. (2020, June 25). *The new generation of ransomware - An in-depth study of Ransomware-as-a-Service*. Retrieved from University of Twente.
- [11] Molina, R., Torabi, S., Sarieddine, K., Bou-Harb, E., Bouguila, N., & Assi, C. (2022). On Ransomware Family Attribution Using Pre-Attack Paranoia Activities. *IEEE Transactions on Network and Service Management*, 19(1), 19–36. <https://doi.org/10.1109/TNSM.2021.3112056>
- [12] Kao, D., & Hsiao, S. (2018). The dynamic analysis of WannaCry ransomware. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*. <https://doi.org/10.23919/icact.2018.8323682>

- [13] *5 most common types of ransomware - crowdstrike*. crowdstrike.com. (2023a, January 30). <https://www.crowdstrike.com/cybersecurity-101/ransomware/types-of-ransomware/>
- [14] Patrizio, A. (2021, July 13). *Malware vs. ransomware: What's the difference?* TechTarget. <https://www.techtarget.com/whatis/feature/Malware-vs-ransomware-Whats-the-difference>
- [15] Slandau. (2022, February 14). *The anatomy of a ransomware attack in 2022; stages, patterns, solutions*. CyberTalk. <https://www.cybertalk.org/2022/02/14/the-anatomy-of-a-ransomware-attack-in-2022/>
- [16] *The anatomy of a ransomware attack - AT&T cybersecurity*. AT&T Business. (2022, November 22). <https://www.business.att.com/learn/articles/what-is-the-anatomy-of-a-ransomware-attack.html>
- [17] *New ransomware strain "cactus" exploits VPN flaws to Infiltrate Networks*. The Hacker News. (2023, March 9). <https://thehackernews.com/2023/05/new-ransomware-strain-cactus-exploits.html>
- [18] Security, P. (2023, March 13). *The changing face of ransomware attacks*. Panda Security Mediacycenter. <https://www.pandasecurity.com/en/mediacycenter/security/changing-face-ransomware/>
- [19] Razaulla, S., Fachkha, C., Markarian, C., Gawanmeh, A., Mansoor, W., Fung, B. C. M., & Assi, C. (2023). The Age of Ransomware: A Survey on the Evolution, Taxonomy, and Research Directions. *IEEE Access*, *11*, 40698–40723. <https://doi.org/10.1109/access.2023.3268535>
- [20] Harford, I. (2021, December 22). *10 common types of malware attacks and how to prevent them: TechTarget*. TechTarget. <https://www.techtarget.com/searchsecurity/tip/10-common-types-of-malware-attacks-and-how-to-prevent-them>
- [21] Asghar, H. J., Zhao, B. Z. H., Ikram, M., Nguyen, G., Kaafar, D., Lamont, S., & Coscia, D. (2022). SoK: Use of Cryptography in Malware Obfuscation. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2212.04008>
- [22] *Virus, malware and ransomware: How they differ*. Nakivo. (2021, May 24). <https://www.nakivo.com/blog/virus-ransomware-and-malware-the-differences-explained/>
- [23] Written by Kaitlyn Graham (2022, October 3). *Top 7 ransomware attack vectors and how to avoid becoming a victim*. bitsight. <https://www.bitsight.com/blog/top-7-ransomware-attack-vectors-and-how-avoid-becoming-victim>
- [24] Palo Alto Networks. (n.d.). *What are ransomware attacks?*. <https://www.paloaltonetworks.com/cyberpedia/ransomware-common-attack-methods>

- [25] Freed, A. M. (2021, November 3). *What are the most common attack vectors for ransomware?*. Cybereason. <https://www.cybereason.com/blog/what-are-the-most-common-attack-vectors-for-ransomware>
- [26] Kelley, D. (2021, September 8). *Top 3 ransomware attack vectors and how to avoid them*. TechTarget. <https://www.techtarget.com/searchsecurity/tip/Top-3-ransomware-attack-vectors-and-how-to-avoid-them>
- [27] *What is Malvertising: Examples & how it differs from AD malware: Imperva*. Learning Center. (2019, December 29). <https://www.imperva.com/learn/application-security/malvertising/>
- [28] Kapoor, A., Gupta, A. K., Gupta, R., Tanwar, S., Sharma, G., & Davidson, I. E. (2021). Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions. *Sustainability*, 14(1), 8. <https://doi.org/10.3390/su14010008>
- [29] Lab, S. (2022, January 24). *Leakware-ransomware-hybrid attacks*. Hornetsecurity. <https://www.hornetsecurity.com/en/security-informationen-en/leakware-ransomware-hybrid-attacks/>
- [30] Rees, K. (2022, July 1). *What is leakware? here's what you need to know*. MUO. <https://www.makeuseof.com/what-is-leakware/>
- [31] *What is double extortion ransomware?*. Zscaler. (n.d.). <https://www.zscaler.com/resources/security-terms-glossary/what-is-double-extortion-ransomware>
- [32] Dheepak, G., & Vaishali, D. (2021). A Comprehensive Overview of Machine Learning Algorithms and their Applications. *International Journal of Advanced Research in Science, Communication and Technology*, 12–23. <https://doi.org/10.48175/ijarsct-2301>
- [33] *What is ransomware?*. Trellix. (n.d.). <https://www.trellix.com/en-us/security-awareness/ransomware/what-is-ransomware.html>
- [34] Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly.
- [35] *Introduction to support vector machines (SVM)*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/>
- [36] *Support Vector Machine (SVM) algorithm*. javatpoint. (n.d.). <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
- [37] Panzura. (2023, May 4). *Ransomware Unlocked: Expert Insights on Attack Prevention and Data Resiliency | Panzura Webinar* [Video]. YouTube. <https://www.youtube.com/watch?v=kNs6tsnrZPA>

- [38] Gandhi, R. (2018, July 5). *Support Vector Machine - introduction to machine learning algorithms*. Medium. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- [39] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [40] Ransomware Attacks: - Impact, Symptoms, Working, Preventive Measures and Response. (2020). *International Journal of Engineering and Advanced Technology*, 9(6), 188–191. <https://doi.org/10.35940/ijeat.f1336.089620>
- [41] *1.4. Support Vector Machines*. scikit. (n.d.). <https://scikit-learn.org/stable/modules/svm.html>
- [42] Mishra, M. (2020, September 2). *Convolutional Neural Networks explained*. Medium. <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>
- [43] *Introduction to convolution neural network*. GeeksforGeeks. (n.d.-b). <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
- [44] Rivas, P. (2020). *Deep Learning for Beginners: A beginner's guide to getting up and running with deep learning from scratch using Python*. Packt Publishing Ltd.
- [45] O'Shea, K., & Nash, R. R. (2015). An Introduction to Convolutional Neural Networks. *ArXiv E-prints*. <https://lib-arxiv-008.serverfarm.cornell.edu/pdf/1511.08458.pdf>
- [46] Fan, Z., Li, W., Jiang, Q., Sun, W., Wen, J., & Gao, J. (2021). A Comparative Study of Four Merging Approaches for Regional Precipitation Estimation. *IEEE Access*, 9, 33625–33637. <https://doi.org/10.1109/access.2021.3057057>
- [47] Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.  
Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160, 3-24
- [48] Palenzuela, F., Shaffer, M., Ennis, M., Gorski, J., McGrew, D. S., Yowler, D., White, D. K., Holbrook, L., Yakopcic, C., & Taha, T. M. (2016). Multilayer perceptron algorithms for cyberattack detection. *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*. <https://doi.org/10.1109/naecon.2016.7856806>
- [49] Warbhe, S. (2021, March 11). *Industry use cases of neural networks*. Medium. <https://surajwarbhe777.medium.com/industry-use-cases-of-neural-networks-e494920fc939>
- [50] R, V., Ganesh, H. B., Poornachandran, P., Kumar, M. A., & Soman, K. P. (2018). *Deep-Net: Deep Neural Network for Cyber Security Use Cases*. [Preprint]. arXiv, cs.LG,

1812.03519.

- [51] Posey, B. (2019, April 3). How has ransomware recovery changed in recent years. <https://www.techtarget.com/searchdisasterrecovery/answer/How-has-ransomware-recovery-changed-in-recent-years>
- [52] Cavalancia, N. (2019, January 21). *Protect backup from ransomware attacks and recover safely: TechTarget*. TechTarget. <https://www.techtarget.com/searchdatabackup/tip/Protect-backup-from-ransomware-attacks-and-recover-safely>
- [53] Masum, M., Faruk, J. H., Shahriar, H., Qian, K., Lo, D., & Adnan, M. I. (2022). Ransomware Classification and Detection With Machine Learning Algorithms. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. <https://doi.org/10.1109/ccwc54503.2022.9720869>
- [54] *How deep learning can be used for malware detection*. reciprocity. (2022, June 23). <https://reciprocity.com/blog/deep-learning-can-be-used-for-malware-detection/>
- [55] Kalita, D. (2022, March 11). *A brief overview of recurrent neural networks (RNN)*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/>
- [56] Brownlee, J. (2021a, July 6). *A gentle introduction to long short-term memory networks by the experts*. MachineLearningMastery. <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>
- [57] A guide to long short term memory (LSTM) networks. (n.d.). <https://www.knowledgehut.com/blog/web-development/long-short-term-memory>
- [58] Lindemann, B., Müller, T. D., Vietz, H., Jazdi, N., & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99, 650–655. <https://doi.org/10.1016/j.procir.2021.03.088>
- [59] SOHEIL, A., ABD RAHMAN, N. A., & OSMAN, H. (2018). A REVIEW OF LATEST WANNACRY RANSOMWARE: ACTIONS AND PREVENTIONS. *Journal of Engineering Science and Technology*. [https://jestec.taylors.edu.my/SpecialIssueICCSIT2018/ICCSIT18\\_03.pdf](https://jestec.taylors.edu.my/SpecialIssueICCSIT2018/ICCSIT18_03.pdf)
- [60] Sgandurra, Daniele & Muñoz-González, Luis & Mohsen, Rabih & Lupu, Emil. (2016). Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection.
- [61] Fernando, D. W., Komninos, N., & Chen, T. C. (2020). A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques. *Iot*, 1(2), 551–604. <https://doi.org/10.3390/iot1020030>
- [62] Zahoor, U., Khan, A., Rajarajan, M., Khan, S. H., Asam, M., & Jamal, T. (2022). Ransomware detection using deep learning based unsupervised feature extraction and

a cost sensitive Pareto Ensemble classifier. *Scientific Reports*, 12(1).  
<https://doi.org/10.1038/s41598-022-19443-7>

- [63] Urooj, U., Al-Rimy, B. a. S., Zainal, A., Ghaleb, F. A., & Rassam, M. A. (2021). Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. *Applied Sciences*, 12(1), 172.  
<https://doi.org/10.3390/app12010172>
- [64] Zahoor, U., Rajarajan, M., Pan, Z., & Khan, A. (2022). Zero-day Ransomware Attack Detection using Deep Contractive Autoencoder and Voting based Ensemble Classifier. *Applied Intelligence*, 52(12), 13941–13960. <https://doi.org/10.1007/s10489-022-03244-6>
- [65] Khan, F., Ncube, C., Kumar, R. L., Kadry, S., & Nam, Y. (2020). A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning. *IEEE Access*, 8, 119710–119719. <https://doi.org/10.1109/access.2020.3003785>
- [66] 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. doi:10.1109/ikt.2013.6620049
- [67] Ryan, M. (2021). *Ransomware Revolution: The Rise of a Prodigious Cyber Threat*. 85- Springer(2021).Springer Nature.
- [68] DiMaggio, J. (2022). *The Art of Cyberwarfare: An Investigator's Guide to Espionage, Ransomware, and Organized Cybercrime*. No Starch Press.
- [69] Guerrero, P. (2023, April 3). *How ransomware will use AI to target it and OT Systems*. CyberClan. <https://cyberclan.com/us/knowledge/how-ransomware-will-use-ai-to-target-it-and-ot-systems/>
- [70] Fritsch, L., Jaber, A., & Yazidi, A. (2022). An Overview of Artificial Intelligence Used in Malware. In E. Zouganeli, A. Yazidi, G. Mello, & P. Lind (Eds.), *Nordic Artificial Intelligence Research and Development* (pp. 41-51). Cham: Springer International Publishing.
- [71] Awati, R. (2022, March 10). *What are metamorphic and polymorphic malware?*. Security. <https://www.techtarget.com/searchsecurity/definition/metamorphic-and-polymorphic-malware>
- [72] Alvarez, R. (2016, June 7). *On-demand polymorphic code in Ransomware*. Fortinet Blog. <https://www.fortinet.com/blog/threat-research/real-time-polymorphic-code-in-ransomware>
- [73] Crane, C. (2021, June 11). *Polymorphic malware and metamorphic malware: What you need to know*. Hashed Out by The SSL StoreTM. <https://www.thesslstore.com/blog/polymorphic-malware-and-metamorphic-malware-what-you-need-to-know/>

- [74] Rissgrouphub. (n.d.). *Rissgrouphub/RANSOMWAREDATASET2016: Ransomware dataset for arXiv:1609.03020*. GitHub.  
<https://github.com/rissgrouphub/ransomwaredataset2016?fbclid=IwAR3AncK0QtTbG81cC5IJIPmtrsLkNsKYJbXNCPYOfZCdO7-0Q7S86nbQgfM>
- [75] Bae, S. H., Lee, G. B., & Im, E. G. (2019). Ransomware detection using machine learning algorithms. *Concurrency and Computation: Practice and Experience*, 32(18).  
<https://doi.org/10.1002/cpe.5422>
- [76] Maniath, S., Ashok, A., Poornachandran, P., Sujadevi, V. G., Sankar, A. U. P., & Jan, S. (2017). *Deep learning LSTM based ransomware detection*.  
<https://doi.org/10.1109/rdcape.2017.8358312>
- [77] Thangapandian, V. (2022). Machine Learning in Automated Detection of Ransomware: Scope, Benefits and Challenges. In *Springer eBooks* (pp. 345–372).  
[https://doi.org/10.1007/978-3-030-93453-8\\_15](https://doi.org/10.1007/978-3-030-93453-8_15)
- [78] *Ransomware attacks: Why email is still the most common delivery method*. Agari. (2023, January 26). <https://www.agari.com/blog/ransomware-attacks-why-email-still-most-common-delivery-method>
- [79] Varsanov, E. (2015, June 2). *Remove locker v1.7 ransomware and restore encrypted files*. Updated. <https://virusresearch.org/remove-locker-v1-7-ransomware-and-restore-encrypted-files/>
- [80] Li, X., Wu, S., Li, X., Yuan, H., & Zhao, D. (2020). Particle Swarm Optimization-Support Vector Machine Model for Machinery Fault Diagnoses in High-Voltage Circuit Breakers. *Chinese Journal of Mechanical Engineering*, 33(1).  
<https://doi.org/10.1186/s10033-019-0428-5>
- [81] Kibria, Hafsa & Matin, Abdul. (2022). The Severity Prediction of The Binary And Multi-Class Cardiovascular Disease - A Machine Learning-Based Fusion Approach. 10.48550/arXiv.2203.04921.
- [82] Afan, Haitham & Ibrahem Ahmed Osman, Ahmedbahaaldin & Essam, Yusuf & Najah, Al-Mahfoodh & Huang, Yuk & Kisi, Ozgur & Sherif, Mohsen & Chau, Kwok & El-Shafie, Ahmed. (2021). Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques. *Engineering Applications of Computational Fluid Mechanics*. 15. 1420-1439. 10.1080/19942060.2021.1974093.
- [83] Maigida, A. K., Abdulhamid, S. M., Olalere, M., Alhassan, J. K., Chiroma, H., & Dada, E. G. (2019). Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms. *Journal of Reliable Intelligent Environments*, 5(2), 67–89. <https://doi.org/10.1007/s40860-019-00080-3>