applied sciences

MDPI

Review

Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions

Umara Urooj [1,*], Bander Ali Saleh Al-rimy [1], Anazida Zainal [1], Fuad A. Ghaleb [1] and Murad A. Rassam [2,3]

[1] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru 81300, Johor, Malaysia; bander@utm.my (B.A.S.A.-r.); anazida@utm.my (A.Z.); abdulgaleel@utm.my (F.A.G.)
[2] Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; muradrassam@ieee.org
[3] Faculty of Engineering and Information Technology, Taiz University, Taiz 6803, Yemen
* Correspondence: umaraurooj@gmail.com

Abstract: Ransomware is an ill-famed malware that has received recognition because of its lethal and irrevocable effects on its victims. The irreparable loss caused due to ransomware requires the timely detection of these attacks. Several studies including surveys and reviews are conducted on the evolution, taxonomy, trends, threats, and countermeasures of ransomware. Some of these studies were specifically dedicated to IoT and android platforms. However, there is not a single study in the available literature that addresses the significance of dynamic analysis for the ransomware detection studies for all the targeted platforms. This study also provides the information about the datasets collection from its sources, which were utilized in the ransomware detection studies of the diverse platforms. This study is also distinct in terms of providing a survey about the ransomware detection studies utilizing machine learning, deep learning, and blend of both techniques while capitalizing on the advantages of dynamic analysis for the ransomware detection. The presented work considers the ransomware detection studies conducted from 2019 to 2021. This study provides an ample list of future directions which will pave the way for future research.

Keywords: machine learning; deep learning; ransomware; ransomware analysis; dynamic analysis; ransomware detection; Internet of Things (IoT); cloud; encryption

Citation: Urooj, U.; Al-rimy, B.A.S.; Zainal, A.; Ghaleb, F.A.; Rassam, M.A. Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. *Appl. Sci.* **2022**, *12*, 172. https://doi.org/10.3390/app12010172

Academic Editor: Agostino Forestiero

Received: 26 November 2021
Accepted: 21 December 2021
Published: 24 December 2021

1. Introduction

Cybersecurity has been one of the major concerns since the development of computers and networking technologies. As the criminals are getting smart and introducing new threats, considerable research has been carried out to develop the counteractions to save individuals and organizations from such sabotage [1,2]. Crypto-virology and crypto-viruses concepts were introduced back in 1996. This concept was later named as crypto-ransomware. The study shed light on positive as well as negative usage of cryptography, in which it was addressed as a double edge sword. This study also created an offensive use of cryptography, by opening a new field of malicious use of cryptography, which provided a base for the many viruses utilizing cryptography to cause damage [3].

Malware is a malicious program aimed at gathering sensitive information, causing disturbance or destruction to single or multiple users [4,5]. It was firstly observed in the late 1970s. It usually gets access to legitimate resources to cause trouble to perform the normal actions. Ransomware is a form of malware that infects the user by encrypting data without user permission. It restricts the legitimate access to user data. It stops users' access to their own resources, i.e., data. The irreversible effect of a ransomware attack makes it distinct from other malwares. Once encryption is achieved there is no other way to decrypt the user files except for by using the decryption key. To make the data decrypted, attackers ask for money in an untraceable currency, i.e., bitcoin. Ransomware attacks are not only affecting the individuals but organizations to gain more money. Attackers take advantage

of untraceable money and irreversible effect of ransomware attack. The victim is threatened that his data will be used or lost, or that sensitive information such as search history will be exposed [6].

In most of the cases the data is disclosed. A study said about 70% of ransomware attacks disclosed the victim's data. The occurrence of ransomware attacks leads to the inaccessibility of data either temporarily or permanently, causing loss to many users and business organizations. Ransomware attacks are spreading due to their lethal effects and monetary incentives [7]. This malware is also tricky to handle as it uses the lethal combination of two strategies to attack. Some of ransomware attacks uses asymmetric cryptography for encryption along with deleting the recovery points and shadow copies [8]. One of the prominent features of ransomware is that it looks like a benign program, making it hard to distinguish ransomware code from legitimate encryption applications [9].

Ransomware attacks can cause trouble to distributed environment to stop smooth work among heterogenous data centers. These systems have complex structures of corpora and algorithms. These environments, i.e., data centers contain huge magnitude of data and can pay ransom to avoid exploitation of data and reputation [10–12].

The reason for a successful attack is that the victim organizations want to get their valuable data back or fear of losing potential users usually goes for paying the ransom. Other victims include users who are unaware of security breaches and wants to get data back. So, the victims with little knowledge usually go for paying the ransom. This tarnished attack is growing with every passing day causing data lose or money lose to users and organizations [13]. Apart from development of many studies to address this attack, the attack ratio is still growing due to development of new variants, easy to use kits (RaaS), and obfuscation techniques [14,15].

In the literature, three different analyses have been carried out for ransomware detection, which include static, dynamic, and hybrid. Each analysis approach has its own benefits and limitations. Dynamic analysis plays up the accuracy of detection by executing the samples [16,17].

Machine learning and deep learning have influences on all the aspects of life. These technologies have serval applications in every domain due to the ability of decision making [18]. These also find several applications in the computing and growing in cybersecurity. Advance attacks and threat detection became easier in less time due to use of these potential technologies [19,20]. Deep learning is best to detect the patterns of an undergoing mechanism. Due to it its pattern recognition ability, it finds application in many domains, i.e., medical, security, AI, and entertainment [20,21]. The machine and deep learning technologies are booming technologies and are vastly used in the advanced research of cybersecurity. These technologies should also be used in the field of ransomware detection. Both technologies played effective roles in pattern identification. A numerous of malware and ransomware are detected by utilizing the machine learning, deep learning, or both [22,23].

A user can only be saved from being a victim of ransomware attack if the detection takes place before the encryption process starts. So, detection plays a vital role in protection from the ransomware attacks. To build an effective solution, an in-depth study of all the available solutions of ransomware detection is required. Hence, there is a need of a state of art survey that provides an overview of existing studies in the field of ransomware detection.

So far, a bunch of surveys and reviews are available which addresses the ransomware evolution, and defense mechanisms. However, each of these studies are limited in context of considered parameters. There is not a single study that considered the dynamic analysis for ransomware detection studies utilizing the available methods of machine learning and deep learning for all the available platforms. This is the first study presenting the more generalized overview of the analysis and detection studies while considering the multitude platforms been attacked. This study also provided the dataset utilized to carry out dynamic analysis for the ransomware detection studies utilizing machine and deep learning methods. To the best of our knowledge, there is no survey paper that combines

the datasets, analysis, detection techniques in an organized manner and give an overview of the studies conducted over period of 2019 to 2021.

The presented study discussed the taxonomy related to ransomware detection studies while presenting some new concepts. This study considered the solutions provided by machine and deep learning to detect and prevent the occurrence of ransomware attack on all the available platforms. The latest ransomware detection studies conducted in 2019 to 2021, performed the dynamic analysis were grouped together to facilitate the future research on ransomware analysis and detection.

To this end, this paper strives to fill this gap and provide a comprehensive overview to facilitate future research. This study will assist the researchers and developers who wish to utilize machine or deep learning methods for the detection of crypto ransomware. This survey will help them in their endeavors to find the adequate solutions for each category.

1.1. Paper Contribution

The contributions of this survey are listed below:

i. This survey presented a brief taxonomy while outlining the ransomware attack detection studies from 2019 to 2021 with the focus on dynamic analysis for different platforms.
ii. Presented a collection of datasets, their sources and analysis tools been utilized to carry out the dynamic analysis, to train and test the ransomware detection systems developed using machine and deep learning techniques.
iii. Extensive overview of ransomware detection studies utilizing machine and deep learning and categorized the studies with respect to encryption process timelines.
iv. Extraction of the list of research directions that need to be addressed by future researcher.

1.2. Paper Ogranization

The structure of the study is organized as follows. In Section 2, competing surveys are presented. Section 3 provides an overview of ransomware taxonomy. It contains different aspects of ransomware which includes, parameters reflecting the occurrence of ransomware attack, ransomware attacking phases, ransomware attack behavior, zero-day attacks, machine learning, deep learning for ransomware detection, adversarial machine learning, and ransomware detection approaches. Section 4 provides the overview of the ransomware compromised platforms from 2019 to 2021. Section 5 contains novel and detailed datasets of ransomware detection studies. Section 6 analyzes the ransomware detection literature in context of the use of machine and deep learning techniques and an extensive review on the ransomware analysis and detection studies are presented. Section 7 presents the research directions for future research. Section 8 concludes the presented survey study. In Figure 1, the ransomware detection taxonomy is drawn.

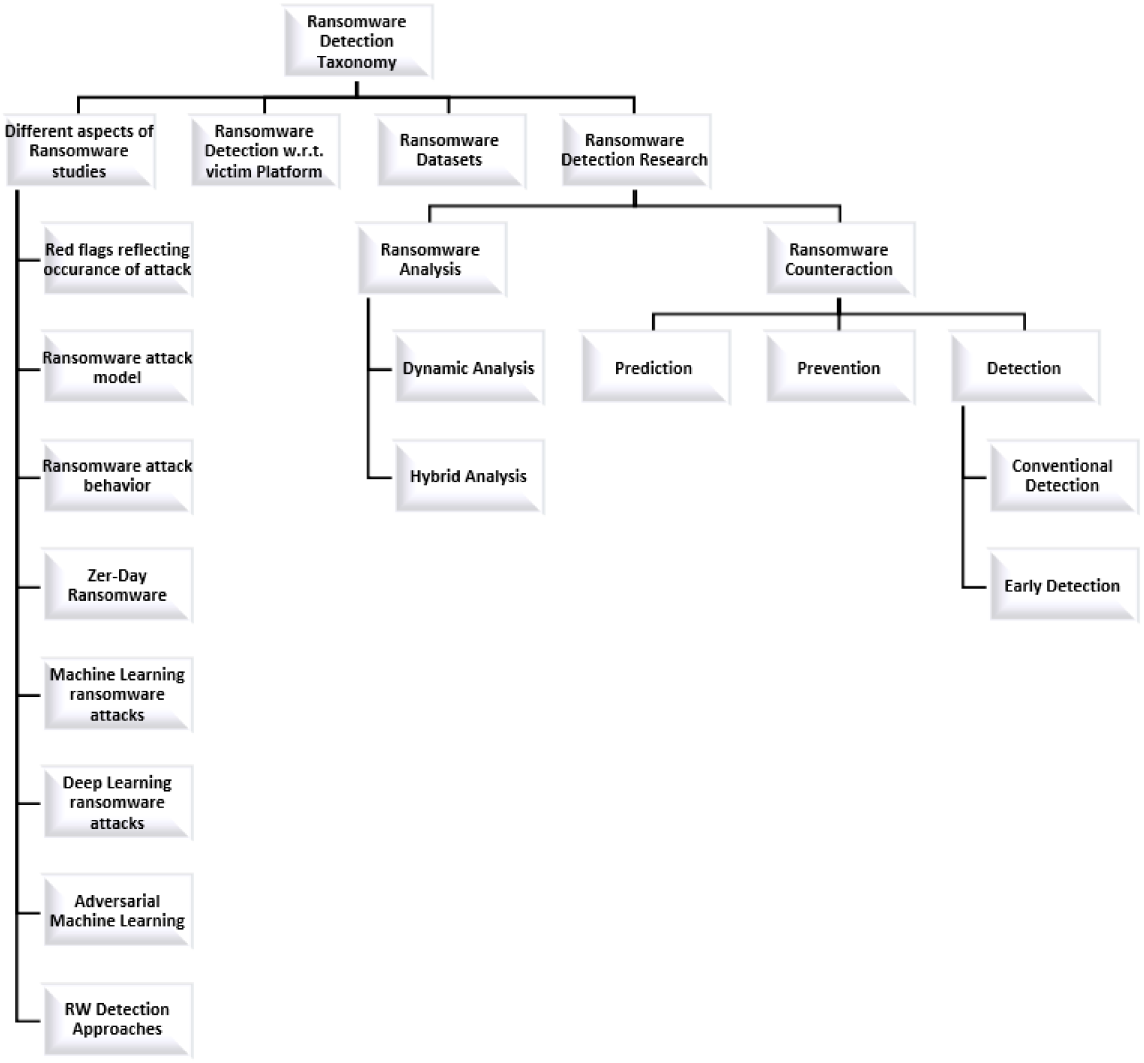

Figure 1. Taxonomy of ransomware detection studies utilizing ML and DL.

2. Competing Surveys

Ransomware is a prominent and evolving area of research. So, different studies including reviews, survey, and systematic literature reviews (SLR) have been conducted to help the researchers and developers to know what is happening in this area. The following discussion will highlight the parameters and aspects discussed in the available literature.

A survey on the ransomware studies was presented in [24]. This study presented a brief review about ransomware classification, types, enablers, infection vectors, platforms targeted, and life cycle of ransomware attack. Ransomware studies were classified into prediction, prevention, and detection. This study presented a detailed taxonomy about the ransomware studies.

A comprehensive review of different ransomware studies was given in [25]. The study suggested some parameters to avoid ransomware attacks. A situational awareness model was presented which helped in decision making to avoid ransomware attack at the early stage. Available literature for ransomware was classified according to review, proposal, testing, detection, and prevention studies. It presented a summary by considering a few parameters, dataset, and tools utilized in the ransomware studies.

A windows platform based ransomware attacks detection survey was conducted in [26]. The presented study reviewed 40 papers and identified ransomware attacks

patterns, methodologies, and attacking mechanism. Ransomware attacks for the windows-based platform were identified along their prominent features.

In [27], a summary of ransomware detection techniques was presented. Different ransomware detection techniques, their pros and cons, recent research from 2015 to 2018, and type of analysis used for ransomware detection were briefly explained. This study presented the methodologies adopted for the detection, generated results, and limitation of each study.

A review on android ransomware studies, using deep learning techniques was presented in [28]. An overview of deep learning approaches and methods was discussed briefly. A review of eight studies for android platform using deep learning for ransomware detection was presented. The study highlighted the deep learning techniques used for the android platform.

A review about applications of machine and deep learning methods, provided security methods for the IoT devices was presented in [29]. Different threat to IoT devices at different layers were discussed. The roles of machine and deep learning methods were discussed in detail with respect to security on different layers of IoT infrastructure.

The study of [30] performed comprehensive survey about deep learning threats, attacks, and defensive techniques. The study discussed both security and private attacks along with countermeasures to each attack category. The security attacks were further divided into two categories, from which the attacks related to deep learning countermeasures were outlined.

In [31], the authors presented a survey of deep learning methods used for cyber security applications. Different attack platforms were considered, which utilized the deep learning techniques. The study discussed the utilized datasets and metrics for deep learning evaluation. Documented uses of deep learning methods were also reviewed in this study.

In [32], the effectiveness of ransomware detection using machine learning methods applied to CICAndMal2017 dataset was examined by conducting two experiments. In first experiment the classifiers were trained on a single dataset containing different types of ransomwares. While, in the second experiment, different classifiers were trained on datasets of ten ransomware families distinctly.

A survey of detection of ransomware using machine learning and deep learning algorithms was presented in [33]. The study focused on finding the weakness in machine learning approaches and ways to strength them. It was suggested in this work, a blend of machine and deep learning can help to find zero-day ransomware attacks. Each research paper was evaluated in context of limitation and improvement. The study also discussed the population drift concept in ransomware.

In [34], a survey was presented that depicted the complete picture on the ransomware and ransomware defense research by considering the diversity of attacked platforms. This survey considered studies until 2020. The focus of this study was to overview the ransomware evolution, attacks, analysis, detection, and defensive mechanisms while considering the different platforms.

A survey about analysis and detection of android ransomware attack was presented in [35]. The study primarily focused on ransomware attacks on android operating system. The survey encompassed the studies from 2015 to 2020 that were related to ransomware analysis and detection methods for the android platform.

A brief review about different attacks and the use of machine and deep learning techniques to prevent the cybercrimes was discussed in [36]. Phishing, social network spam, malware, ransomware, network intrusion detection studies utilized the machine and deep learning technologies were discussed.

The survey presented in [37] discussed the use of deep learning technology in cybersecurity domain. A brief review about the deep learning solutions was described. The survey was outlined by discussing the studies of malware and android malware detection, traffic and binary analysis, spam and phishing detection, and intrusion detection.

A study about ransomware evolution, prevention, and mitigation in the domain of IoT was presented in [38]. Ransomware attacks on the IoT devices, its propagation, prevention and mitigation approaches, and challenges were discussed comprehensively.

A systematic literature review for windows operating system was presented in [39]. This study discussed different types of ransomwares along with type of encryption used by each ransomware, infection methods, attack methodologies, impacts, vulnerabilities, available mitigation, and prevention from the attacks.

A brief survey about ransomware anti-analysis and evasion techniques was presented in [40]. Each of the anti-analysis or evasion techniques were explored precisely. Ransomware attacked platforms, types of ransomwares along with effects of the attacks were also briefly explained.

A study presented in [41] summarized the study related to android systems. The detection and protection from ransomware attacks and comparison of those schemes was also presented. A comparison for the analysis results were discussed briefly. The study discussed the different type of analysis carried out in android ransomware studies. The available protection technologies used to prevent android ransomware attacks were described along with comparison of these schemes.

A system literature review was presented in [42] for windows and android systems, which summarized the ransomware attacks and protection surveys. The study presented the ransomware attacks taxonomy. Paper also discussed the different ransomware detection techniques, their tackled problems, solutions presented to the problems and limitations of those solutions.

The study in [43] presented a survey about deep learning methods in the field of cybersecurity. The study highlighted the security issues faced by the deep learning models. The roles of deep learning and its architectures were briefly discussed. Deep learning studies which considered protection techniques were explored.

A comprehensive survey [44] of ransomware detection studies including analysis and synthesis was conducted. The survey highlighted the role of machine learning algorithms for the defensive solutions against ransomware attack. This study also presented brief dataset of ransomware detection studies. Some aspects of deep learning and bigdata were also included in the study.

A survey [45] of ransomware detection techniques using machine learning, deep learning, and artificial intelligence was conducted. This paper primarily focused on the ransomware detection by using the natural language processing techniques. Different aspects of ransomware studies were discussed in the survey. Different analysis carried out for different detection techniques were explained.

A brief survey about deep learning techniques was presented in [46]. The concepts involved in deep learning studies were concisely presented. Some of ransomware detection studies utilizing the deep learning were also explored. Deep learning methods and analysis were presented for the different platform. Evaluation metrics for presented study were also listed.

3. Different Aspects of Ransomware Studies

Ransomware studies were increasing as more research is required in this emerging field. In this section different aspects related to ransomware studies are described.

3.1. Red Flags Reflecting Occurrence of Ransomware Attack

There are some behaviors patterns which depict the occurrence of a ransomware attack. Listed actions can be used as a flag to detect the launch of a ransomware attack.

(1) Opening of many files [47].
(2) Structure of input and output streams to a process is different [47].
(3) Many write/overwrite operations [48].
(4) A process calling encryption APIs [49].
(5) Frequent reading and rewriting/deleting requests in a short period of time [50].

(6) Communication with command-and-control server [48].
(7) Change in the user registry keys [51].

3.2. Ransomware Attack Model

Ransomware attacks follow a specific pattern that can be observed in each family and variant of ransomware [52]. In general, ransomware attack is launched in three phases [53]. A successful ransomware attack can be categorized in three phases which are pre-encryption, encryption, and post-encryption. Figure 2, describes the occurrence of a successful ransomware attack.

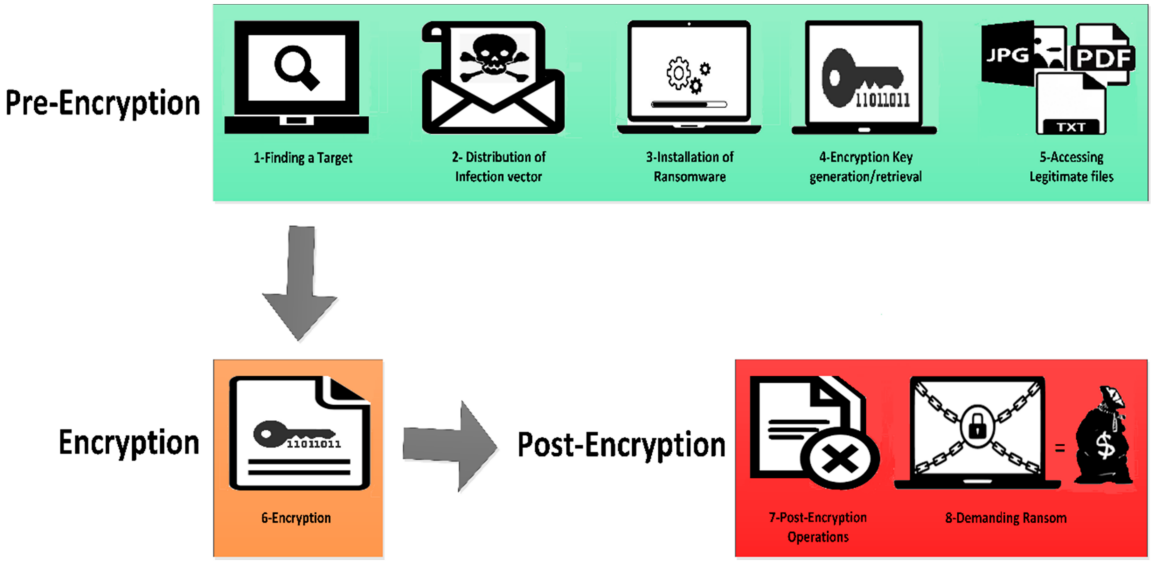

Figure 2. Ransomware attack phases.

(1) Finding a target;
(2) Distribution of the infection vector;
(3) Installation of ransomware;
(4) Encryption key generation and retrieval;
(5) Accessing legitimate files;
(6) Encryption;
(7) Post Encryption operations;
(8) Demanding Ransom.

3.3. Ransomware Attack Behaviour

Ransomware performs a list of operation while encrypting the victim's files. Ransomware behaves in similar ways but differs in payloads [6]. According to the set of operations performed [9] ransomware attack falls in three classes. This behavior was also defined by [14,54] in similar way. Figure 3, elaborated the behavior of each class pictorially.

Class A: Class A ransomware encrypts the original data. It does not create a copy of original data. This class ransomware includes opening, reading, writing, and closing the original file. This class optionally renames the encrypted files.

Class B: A class B ransomware attack occurs while taking original data out of victim's directory. Then encryption is performed on a copy of original data. Encrypted files are placed back to the victim's directory. Encrypted files can have different names from original file.

Class C: A class C ransomware attack reads the original files and independent encrypted files for the original data are created separately. Then original files are deleted to dispose the original content. Overwriting is performed using the move operation.

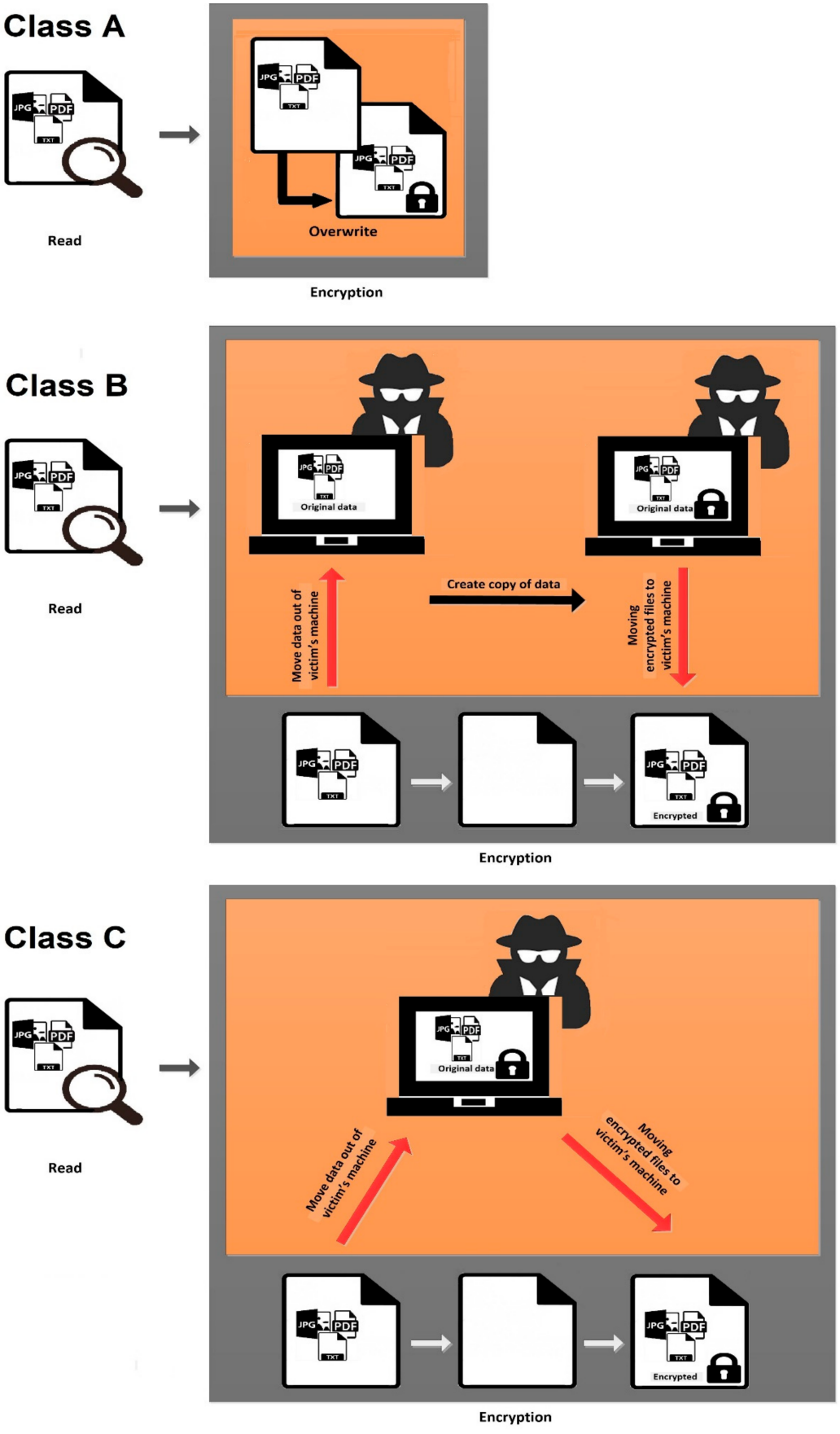

Figure 3. Ransomware attack behavior.

3.4. Zero-Day Ransomware

Zero-day ransomware are those variants which are not discovered previously. No information about them is available till the date they reported first. These ransomwares are detected by the detection system which are well designed to detect the unknown ransomware variants also. Different studies in the literature address the zero-day detection. Some of these are mentioned below.

A system was proposed in [14] which claimed ransomware detection with no data lose. A windows desktop screen was monitored as a parameter to detect ransomware attack. This technique of finding a change in the desktop was the base of detection the previously unknown ransomwares.

Another mobile ransomware prevention technique was proposed in [55] for the android platform. This method was able to detect modified patterns of ransomware. To perform the ransomware detection, this technique monitored the file events by observing the CPU and I/O usage.

A ransomware detection system for the fog layer was presented in [20], that can detect the zero-day ransomware attacks. This study utilized the two deep learning technologies LSTM ad CNN to detect and distinguish the ransomware from good ware. However, this system starts working after the 10 s of launch of an application. This study considered few ransomware families to train the model.

A decoy-based ransomware detection system was proposed in [56]. This system observed the deployed honey files to detect the crypto-ransomware. Whenever a ransomware reaches the honeypots, the detection is triggered. However, this detection system raised high false alarms.

3.5. Machine Learning Ransomware Attacks

Machine learning has a great impact on cybersecurity field. It has numerous applications to detect the intrusion, fraud, malware, and ransomware. Machine learning approaches are effective in detecting the hidden patterns. Machine leaning has numerous application in classifying the malware, ransomware, and benign programs [22].

R-PackDroid, an android based ransomware detection system was presented in [57]. This system performed the static analysis by analyzing the Dalvik bytecode. The Crypto-API calls of applications were examined to extract the features. However, the system was limited in its functionality as it cannot detect the system API calls or encrypted operations that run during the execution.

EldeRan, a machine learning based system was presented in [58]. This system was able to detect the previously unseen ransomware families. This system captured the significant dynamic features, for the ransomware detection and classification. An evaluation of LR, SVM and NB was also performed. The most relevant features were selected using the mutual information criteria while classification was performed using LR. However, the system ran the samples for only 30 s which made the system inaccurate for the ransomware attacks utilizing the obfuscation techniques.

A windows based NetConverse system was developed in [59] for the network ransomware detection. Different machine learning classifiers were selected including MLP, BN, DT, KNN, RF and LMT. These classifiers were compared in term of TP and time required to build a detection model. A network protocol analyzer extracted the features from the network traffic.

In [60], a detection system was proposed which utilized the sequence pattern mining to detect the ransomware. Different classifiers were used including Bagging and MLP, and two machine learning classifiers J48 and RF. Frequent features of ransomware were detected by using the sequence pattern mining. Ransomware logs were built by using the vectors composed of ransomware frequent patterns. This study emphasized on the use of pattern mining techniques for the extraction of best features for the ransomware hunting applications.

3.6. Deep Learning Ransomware Attacks

The deep learning methods are more adaptive in inferring the data [61]. The use of deep learning techniques is increasing and providing good results in the ransomware detection [62]. Deep learning is a robust approach which provides good results in extraction of hidden patterns from the unsupervised data [16].

A memory assisted stochastic dynamic fixed-point arithmetic-based ransomware detection system was proposed in [48]. Random bit streams hold by memory was used to train the deep belief network classifiers. This was an online detection system, presented for the hardware with limited memory and power resources, i.e., IoT and embedded systems. System focused on the accuracy and the speed of detection.

An automated study utilizing the deep learning technique LSTM to detect the ransomware was presented in [63]. The system utilized API calls to observe the behavior of a process. Tensorflow was used to implement the LSTM network. The system was designed to deal with the obfuscation techniques of ransomware. Analysis was carried out for 20 min to deal with the obfuscation techniques.

A detection system was presented in [64], which utilized machine learning classifier SVM and artificial neural network technology MLP. API calls were captured to detect the ransomware existence. MLP outperformed as compared to other shallow networks. MLP performed well in the detection and classification of ransomwares in comparison to the other classical machine learning classifiers while the deep network performed well in comparison to other traditional shallow networks.

A machine and deep learning-based ransomware detection model was presented in [65] for the mobile devices. This preventive study was able to detect the ransomware attack in a minute. This model was able to detect unknown ransomware samples in the real time. Initially, the fed samples were analyzed statistically to check if these were benign or malicious, which can allow the obfuscating ransomware to bypass the detection. This system built the DNA profile for each ransomware family. It compared the fed sample DNA with available DNA profiles. However, due to statistical analyses of samples, model was prone to evasion techniques.

A deep learning method to detect the ransomware was presented in [66]. This detection study utilized the machine and deep learning technologies to detect the ransomware. This detection system collected the payloads and http requests from real-world data. Model was trained on the features extracted from the TCP and UDP traffic. Neural network was built using TensorFlow. This detection model was able to detect the previously unknown ransomware variants. However, this system performed the static analysis, which could be by passed by the obfuscating ransomware.

3.7. Adversarial Machine Learning

Adversarial machine learning is emerging filed to study the machine learning models weaknesses, which make them prone to attacks and find out solutions to stop adversarial manipulation [67]. Adversarial machine learning generates malicious inputs, i.e., ransomware samples to exploit the machine learning detection models. It gathers the information about the detection model and launches the attacks according to the system vulnerabilities [68].

The concept of adversarial machine learning attack was discussed in [67]. A taxonomy for the attacks on machine learning was presented. The study summarized the robust learning techniques and common problems of adversarial learning. The paper discussed the decision timing attack concept on machine learning along with state of art theories and practices.

The study in [69] focused on the adversarial inputs at the test time to evade detection. The study aimed at making the machine learning systems more robust to the adversarial inputs. Adversarial inputs were designed to make model processing difficult at the test time. Different defenses and limitations techniques against the adversarial samples were also presented.

Ensemble analysis of machine learning was used in [70] to mitigate the adversarial evasion attacks on android devices. Adversarial attacks on ransomware samples were performed to evaluate the ensemble analysis. Fabricated inputs of android ransomwares were used to validate the resilience of model against the adversarial attacks.

An injection of the system API calls method was used in [71] to generate the adversarial samples to evade the detection of a state of art detector. The study targeted the use of specific API calls to evade the detection. The vulnerability of detection model was assessed against adversarial samples. Mimicry and random noise were also used to add more calls to weaken the detection.

3.8. Ransomware Detection Approaches

Like malware detection approaches, ransomware detection approaches also categorized into two major classes, i.e., misuse and anomaly based detection approaches [24].

3.8.1. Misuse based Ransomware Detection Approaches

In misuse-based ransomware detection, the signatures of already discovered ransomware are used to draw the conclusion for available threats. It matches the encountered threat with known and available signatures to make decision. Misuse detection is further classified into two categories, i.e., structural or signature and behavioral [24]. A summary of ransomware detection approaches is presented in Table 1.

I. Signature based Detection Approaches

In signature-based detection approach the patterns or codes of already available threats are compared with the under examine code. One of the limitations of this approach includes the outdated samples. New variants are developing with passage of time which required a periodic update to the available ransomware collection. This periodic updating caused overhead [72].

A signature based detection technique was proposed in [8]. This detection approach utilized formal methods for the detection of ransomware in the mobile devices. This detection system inspected the malicious behavior at the bytecode level.

In [57], an android based ransomware detection system was presented named as R-PackDroid. This system ran the static analysis using RF a supervised machine learning classifier to categorize the applications in ransomware, generic malware and trusted by using the system API packages. This system also confirmed the samples uploaded on VirusTotal with low confidence. However, this system was not able to detect the application that load the feature code during execution. The system could not detect the class that are fully encrypted.

A static analysis-based ransomware detection method was proposed in [73]. This system was able to detect the ransomware which can fingerprint the environment. The work was proposed for the binary and multi-level classification and to fill up the gap for dynamic analysis limitations. Static analysis was performed on opcodes, based on text analysis along with machine learning algorithms.

Contextual information was extracted to perform the static analysis in [74]. It was a deep learning-based model which performed the static analysis on N-gram of opcodes. Self-attention mechanism was used on the semantic information to capture the rich context. This system was able to detect the ransomware which could fingerprint environment. However, this approach did not ensure the accuracy due to hard handling of advanced packing techniques adopted by the new variants of ransomwares.

II. Behavioral based Detection Approaches

In the behavioral based detection approach the under-examination samples are executed in the controlled environment to monitor its behavior. The detection is performed in accordance with the behavior, shown by the subjected sample(s).

The study in [9], described an early warning detection system which monitored the ransomware behavior on the user files. This system was run for 20 min which made it

unsuitable for the ransomware detection due to long time. This detection system was also not able to detect the zero-day attacks.

In [49], a real time detection approach was presented which provided early detection. A study was carried out on locky ransomware. A decoy file technique along with process monitoring was used to implement the detection system. A depth first search and recursive approaches were considered for the file traversing. Decoy files were placed in every directory. When a malicious process operated the decoy file, process monitoring module monitored the process. If malicious activity was observed, the process was referred to the user for decision about execution continuation.

A ransomware detection approach called Self Defensible SSD was proposed in [50] utilized storage devices for detection purpose. Storage access activity was transparently monitored to detect the ransomware and once a ransomware was detected, the garbage collector performed the backup work to retain the original data. This approach could perform the recovery of original data without the decryption key. Ransomwares were detected using the frequent read and rewrite/delete requests in the short period of time. This solution was placed inside SSD to perform the transparent monitoring.

A honeypot technique was used in [75] to generate the alarm for ransomware detection. Honeypot folders were created and monitored for the changes occurred. Two methods, Microsoft File Server Resource Manager and EventSentry, were utilized to perform the detection and monitoring. A ransomware attack was monitored on Microsoft windows network. However, this system did not prevent the launch of a ransomware attack. This technique was also not reliable as there was no guarantee the honeypot folders will always be accessed by the attack. It acted like an alarm so that the administrator disabled the user account and shut down the system as final solution to stop the attack.

In [76], an analysis of Cryptowall ransomware in a network environment was presented. Honeypot and maltester were used to carry out the dynamic analysis. All the URLs in the networks were captured by using the honeypot. However, this system was limited in its function, i.e., this approach stopped the analysis to check the target file system. This system identified the infected proxy machines.

III. Hybrid Detection Approaches

A hybrid-based approach takes the advantage of both misuse based and behavioral based detection approaches. This ransomware detection approaches detect the ransomware by combining the use of signature and behavioral based detection approaches [77].

Heldroid [72], a detection technique for android mobile systems used the misuse based approach to perform the detection. Heldroid analyzed the samples dynamically and statistically, which made this approach hybrid. It was also a state-of-art approach for unknown ransomware samples. Both static and dynamic analysis were carried out to categorize the samples into ransomware, goodware, and scareware.

A deep learning based automated ransomware detection tool was proposed in [78]. This detection tool was introduced for the email filtration. It separated the emails with suspicious downloads. Both static and dynamic analysis were carried out to gather the information about ransomware activity. The proactive monitoring system monitored (PMS) the downloads to identify the ransomware attack.

An early detection system named PEDA was proposed in [79] for the ransomware detection. Two level detection was performed with help of hybrid analysis. System calls were collected from the samples after analyzing the samples for 30 s in cuckoo sandbox. This system was also able to detect zero-day attacks. However, this system was not able to detect the ransomware that used its own encryption code.

A hybrid analysis was carried out in [80] to perform the feature analysis. Two different datasets were used to perform static and dynamic analysis. Ransomwares were detected by performing the feature engineering technique to extract the static and dynamic features. Deep convolutional neural networks and machine learning were used to detect the ransomware.

3.8.2. Anomaly based Ransomware Detection Approaches

In anomaly detection the normal behavior is modelled to build a normal profile. This normal profile is always different than the profile built from the anomalies [24,81]. This detection is performed by building a profile with use of benign applications. Any deviation to normal profile is considered as anomaly.

Table 1. Ransomware detection approaches.

Ransomware Detection Approaches					
Sr#	**Study**	**Misuse Based**			**Anomaly Based**
		Signature	**Behavioral**	**Hybrid**	
1	[72]			✓	
2	[9]		✓		
3	[8]	✓			
4	[75]		✓		
5	[82]				✓
6	[78]			✓	
7	[49]		✓		
8	[83]				✓
9	[50]		✓		
10	[57]	✓			
11	[79]			✓	
12	[76]		✓		
13	[84]				✓
14	[73]	✓			
15	[80]			✓	
16	[85]				✓
17	[74]	✓			

The authors in [82] developed the new ransomware variants to describe a dynamic ransomware detection system. A local virtual client server environment was developed and named as File tracker. This was a behavioral based system performed detection in two modes, i.e., user and kernel modes. Normal and Abnormal access profiles were built by using aggregated data from all the 50 clients. Suspicious features were classified into three categories with respect to subject, process, and impact of the attack. Whenever a suspicious feature was observed the user and server were notified. This system possessed certain limitations as it can detect the existing and new ransomware variants with a loss of up to 20 files. Delays were also observed due to client server architecture.

An anomaly based detection and prevention model for ransomware was developed in [83] which performed detection in four phases. Early detection model was developed by carrying out study on the unstructured data of petya and WannaCry logs, stored in an institution EcuCERT. A corpus was created consisted of the most important features in the log files. Corpus contained the behavior pattern of vulnerabilities found in Microsoft windows systems. The study explained the features that best represent the vulnerabilities in the system by using the user log files. Study utilized big data analysis tool with machine learning to identify behavior patterns of ransomware.

A behavioral and anomaly based ransomware detection model was presented in [84]. This model was developed by the integration of two modules, one dealt with behavioral data and other dealt with anomaly data. The anomaly detection module was responsible to

find zero-day ransomware attacks. The anomaly module was built by using the legitimate programs. Any deviation from the normal profile resulted in triggering the alarm.

A deep learning-based ransomware detector DeepRan [85] utilized the anomaly concept to detect the ransomware attacks. This detection model monitored the abnormal activity of infected host in a network environment of bare metal server. The normal profile for the model was developed by collecting the normal host logs. Model was developed by using the deep learning classifier BiLSTM-CRF to classify the normal and infected host activities.

4. Ransomware Detection Studies w.r.t Compromised Platforms

In a report by [86], it was highlighted that not only PCs but many other devices encountered ransomware attacks. These devices included IoT devices, i.e., wearables and smart TVs, mobile devices, fog layer, and cloud-based systems. Different studies presented the ransomware attacks on different computing platform. A comprehensive overview of the available studies is presented below:

(1) Ransomware attacks on PC and smartphones;
(2) Ransomware attacks on IoT ecosystem (Cloud, Fog, IoT, and Ransomwear);
(3) Ransomware attacks on network.

The Table 2 summarizes the ransomware attacks detection studies developed for different platforms conducted in 2019 to 2021.

Table 2. Ransomware detection studies w.r.t. compromised platform.

Ransomware Detection Studies w.r.t. Victim Platform							
Sr#	**Study**	**Desktop and Smartphone**		**IoT Ecosystem**			**Networks**
		PC	**Smartphone**	**Cloud**	**Fog**	**IoT**	
1	[8]		✓				
2	[14]	✓					
3	[48]					✓	
4	[50]	✓					
5	[55]		✓				
6	[57]		✓				
7	[62]				✓		
8	[66]						✓
9	[72]		✓				
10	[75]						✓
11	[76]						✓
12	[82]						✓
13	[83]	✓					
14	[87]	✓					
15	[88]					✓	
16	[89]					✓	
17	[90]			✓			
18	[91]						✓

4.1. Ransomware Attacks on PC and Smartphones

Number of personal computers and its usage is growing every day exponentially. The outdated computers and anti-viruses make the launch of ransomware attack easier. Different operating systems of desktop computers are being targeted [92]. Mobile ransomwares,

attack the mobile platforms. These ransomwares attacks are launched when the mobile user download and install a trojan or a fake app. Fake apps spread via untrusted third-party app stores. Mobile screen gets locked, and all the mobile data gets encrypted including contact lists and the user is asked to pay the ransom. Victims are threatened with the loss of their mobile data, sharing of personal information, and browsing history to their contact lists. Android.Lockdroid.E is one of the example of mobile ransomware [57,86].

A desktop ransomware detection approach was proposed in [50]. This system, named Self Defensible SSD, utilized the storage to detect the ransomware. This system was also able to provide backup by using the garbage collector which hold the original data. This system performed detection and backup without the use of decryption key.

A windows based ransomware detection system was developed in [83]. This model was able to detect the ransomware in the early stages. The system built a normal profile by using the unstructured data of the log files from an institution. This system performed detection and prevention of ransomware attacks by using the machine learning algorithm.

UNVEIL [14], a detection system was proposed for the windows platform. This system performed ransomware detection by using the desktop screenshots and calculating the similarity score for them. System focused on the desktop screen to detect the unknown ransomwares. This system operated in the kernel and worked with file system drivers.

In [72], an android mobile based detection approach HELDROID was presented. This approach also detected the zero-day ransomware. However, this approach is limited due to dependency on the training language of detection model. It required 30 min to train the model for new language.

R-PackDroid [57], an android based ransomware detection system was presented which labeled the program into ransomware, malware, or trusted application. A supervised ML classifier was used to classify the subjected program into benign and ransomware. However, this system was not able to perform the detection during execution.

A technique based on formal methods for the detection of mobile ransomware in the smartphone devices was proposed in [8]. The apps running as android ransomwares were detected and removed from the device. It inspected the malicious behavior at bytecode level by employing the full static approach. This technique lacked in disassembling all selected samples which was limitations of this work. To solve the disassembling problem lower number of morphed samples were utilized. Samples used were taken from top 10 most popular families.

A ransomware prevention technique was developed in [55]. This technique performed detection by collecting data from the processors, I/O, and storage devices. The android processes and directories were monitored to perform the detection. This technique was implemented at kernel level hence imposed less overhead. User was able to decide the processing or termination of a suspicious process.

A two stage ransomware detection model for windows platform was developed in [88]. First stage is developed using Markov and the second stage by using RF. Sequential properties of windows API were collected to build the Markov model. Two separate Markov were built for both benign and ransomware programs. The machine learning classifier random forest was used to classify the program into benign and ransomware.

4.2. Ransomware Attacks on IoT Ecosystem

The Internet of Things (IoT) is a system of interrelated computing devices with limited power and memory constraints. The connected architecture of IoT devices made them more attractive for ransomware attackers, i.e., attackers can attack a large number of victims [48].

A ransomware detection model utilizing CNN, LSTM and OCSVM was proposed in [62]. This detection model was designed to work on the fog layer. The detection was performed by vectorizing the malicious behavior of an application. The OCSVM utilized the generated vectors to find out the family of ransomware while LSTM and CNN performed classification of ransomware.

Authors in [48] proposed a ransomware detection method designed by using the Deep Belief Network (DBN) in hardware. This method utilized the deep learning approach for ransomware detection. This was an online detection method designed for the IoT devices and embedded systems. It was a behavioral detection solution using memory assisted stochastic dynamic fixed-point arithmetic. Method stored the random bits streams in the memory.

A deep learning based ransomware detection model was proposed in [89] for Industrial IoT. The high dimensional data was collected to perform hybrid feature engineering. Classical and variational auto-encoders were used to reduce the data dimension and features extraction. This study focused on the extraction of latent patterns, learning from these patterns, and data dimensionality reduction. DNN with BN was used to make the decision about ransomware and benign.

A ransomware detection model was presented in [90] industrial IoT systems. This model utilized stacked variational autoencoder for unsupervised learning. This study utilized augmentation method to produce training data for fully connected network. Dynamic analysis was performed to capture API calls, file and directory operations, and registry keys. This model utilized fully connected neural network for the detection.

A ransomware detection and prevention system for the backup systems, i.e., Cloud was proposed in [91]. This method utilized the file entropy method for ransomware detection. This method also provided the file restoration of original data if ransomware attack occurred. Machine learning methods were used along with optimal entropy reference value to detect the ransomware attack and its infected files

4.3. Networks Based Ransomware

Due to evolution of more sophisticated ransomwares, packet inspection method is not sufficient to rely on. To detect the ransomware, a network monitoring system should be deployed. A network detection system monitors the network to detect the malicious activity [93].

A network based ransomware detection system was proposed in [75]. Honeypots were created on each workstation to observe the system behavior. The workstations were observed to find the malicious behavior. Once a malicious behavior was observed the network administrator disabled the user account and shut down the system to isolate it with the system. This detection system did not prevent the launch of the attack and lead to lose of data if an attack launched.

In [93], a ransomware detection model was developed. This model performed detection of high survivable ransomware in two stages. Network connection was monitored to find patterns. The user was informed when malicious behavior was observed. The user was able to break the connection by checking the connection address. Connection breaking invoked the unsuccessful key exchange.

A ransomware detection system for the network environment was proposed in [76]. This detection system used honeypots to find out the ransomware attack. The URLs from infected workstation were inspected to detect ransomware. This system worked in on-off fashion. Secondly, the system halted its working when honeypots were checked.

In the study of [66], a deep neural network based ransomware detection model was introduced. It was a network-based detection system that observed the suspicious process by observing the network calls from malicious application. Model was trained by using the generated HTTP requests from malicious application. Both machine and deep learning were used to detect and prevent different variants of ransomware attacks.

In [82] a ransomware detection system was developed for the client server architecture. This system detected the ransomware by comparing the behavior with the normal profile. Normal profile was developed by using the normal activity data from different workstations. The detection was performed by observing the malicious behavior at user and kernel level. Once the malicious behavior was observed the server was notified to isolate the system

5. Ransomware Datasets

Dataset and its source play crucial role in the development of an accurate detection system. Dataset directly relates to the output and accuracy of a detection system. Classifiers trained on invalid data will generate the invalid results, i.e., GIGO. Training and detection are directly related and dependent on the input dataset. Ransomware variants are developed enormously due to the use of polymorphic and metamorphic techniques. Dataset is also very important when developing the adaptive detection system for the obfuscating, polymorphic and metamorphic developed ransomwares. Figure 4, draws the effects of a dataset on the development of an adaptive detection model. The more reliable and trusted dataset will generate the better classifiers [29,41].

Figure 4. Significance of a dataset in a ransomware detection system development.

Different studies used datasets from different repositories. VirusTotal samples are used in most of the ransomware detection studies. Other popular repositories are VirusShare, and theZoo. Some other sources include hybridanalysis.com. Based on number of samples used for the ransomware and benign, many studies used different ratio of both categories. A summary of different dataset repositories and datasets used in the detection studies from 2019 to 2021 for different platforms are shared in the Table 3.

Table 3. Datasets used in ransomware detection studies.

Ransomware Datasets							
Sr#	**Study**	**Tool**	**Platform**	**Dataset Type**	**Family Name**	**Dataset Source**	**# of Samples**
1	[94]	Cuckoo	Desktop Windows	Ransomware		VirusShare VirusTotal	1354
				Benign		Software-informer System 32	1358
2	[20]	Intel Pin 3.2	Desktop Windows	Ransomware		VirusTotal	1000
				Malware		VirusTotal	900
				Benign		Windows 7 system directory	300
3	[89]	Did not mention	Industrial IoT	Ransomware		NA	582
				Benign		Windows application	942
4	DRTHIS [62]	Did not mention Event recorder	Fog Layer	Ransomware		VirusTotal	660
				Benign		NA	219
5	RanSD [80]	Cuckoo	Windows	Static Dataset 3646	Ransomware	VirusTotal VirusShare	1700
					Goodware	Window 7	1946
				Dynamic Dataset 3444	Ransomware	VirusTotal VirusShare	1455
					Goodware	Window 7	1989
6	DeepRan [85]	Log Parser	Networks Bare metal server	Ransomware event logs		PC host logs	17
				Benign event logs		PC host logs	103,330
7	[90]	Did not mentioned tool	Industrial IoT	Ransomware	Sgundara	NA	582
				Benign	Sgundara	NA	942
8	[95]	Cuckoo Sandbox		Ransomware		VirusShare	8152
				Benign		Informer.com	1000

Table 3. *Cont.*

Ransomware Datasets							
Sr#	**Study**	**Tool**	**Platform**	**Dataset Type**	**Family Name**	**Dataset Source**	**# of Samples**
9	RAPPER [81]	Cuckoo sandbox	Windows	Ransomware	Wannacry	NA	NA
					Vipasana		
					Locky		
					Petya		
				Benign		NA	
10	RANDS [96]	Virtual testbed	Windows	Ransomware	AiDS	VirusTotal Malware Blacklist	400
					RaaS		310
					GpCode		800
					CryptoLocker		720
					Archiveus		1500
					CryptoWall		3250
					WinLock		2620
					Reveton		400
				Benign		Website	500
11	[97]	Cuckoo	Windows	Ransomware		VirusTotal Sgandurra theZoo	357
							491
							56
				Benign		Sgandurra	942
12	[7]	Flow exporter Flow controller	IoT	Ransomware	Wannacry		50,537
					Petya		
					BadRabbit		
					PowerGhost		
				Benign		Network Traffic generated by Integrated Clinical Environment (ICE)	100,000
13	[98]	Did not mention	Windows	Ransomware		RISS of ICL machine learning online repository	582
				Goodware			942
14	PEDA [79]	Cuckoo	Windows	Ransomware	Sgandurra	VirusShare VS malware repository theZoo	995
				Goodware		NA	942
15	[99]	Cuckoo	Host in Security Operation Centre	Ransomware attacked logs	WannaCry	Infected System logs	NA
					DBGer		
					Defray		
					Locky		
					Cerber		
					GandCrab		
					nRansom		
				Non attacked logs		Uninfected logs	NA
16	RanStop [100]	Monitoring of micro-architectural events using hardware performance counter	Windows	Ransomware		VirusShare	80
				Goodware		OpenSSL C programs	76
17	[101]	Python script and MATLAB	Networks	Dataset created by the network traffic of Malware Capture Facility Project (MCFP)			
18	[91]	User clint software	Backup Systems	Encrypted files	System files Documents Images Source code Executables Compressed	NA	600

Table 3. *Cont.*

Ransomware Datasets							
Sr#	**Study**	**Tool**	**Platform**	**Dataset Type**	**Family Name**	**Dataset Source**	**# of Samples**
				Normal files	System files Documents Images Source code Executables Compressed	NA	600
19	[102]	Virtual machine, Disco, and process monitoring	Windows	Ransomware		VirusShare	NA
				Benign		NA	NA
20	AIRaD [103]	Sandbox	Windows	Ransomware		VirusTotal	550
				Benign		Windows 10 Open Source Software	540
21	[104]	Cuckoo	Windows	Ransomware 1254	TeslaCrypt	VirusShare VirusTotal	96
					Petya		89
					Pgpcoder		46
					Reveton		50
					CryptoWall		151
					Kollah		73
					Kovter		23
					Citroni		67
					Trojan		82
					CryptLocker		173
					Torrent Locker		108
					Cerber		171
					WannaCry		74
					Dirty Decrypt		51
				Benign 1308		Software.informer System32 of Win7 Pro	NA
22	[105]	Weka and Python code	NA	Ransomware 35015	Archiveus	VirusTotal VirusShare	1500
					CryptoLocker		1720
					AiDS		4000
					RaaS		1300
					Zeus		1500
					Locky		2000
					GpCode		8000
					CryptoWall		3250
					Crysis		1320
					WinLock		3620
					WannaCry		1300
					Sopra		1570
					Reveton		2400
					Cerber		1535
				Malware			500

Table 3. *Cont.*

Ransomware Datasets							
Sr#	**Study**	**Tool**	**Platform**	**Dataset Type**	**Family Name**	**Dataset Source**	**# of Samples**
				Goodware		Developed with Software	500
23	[106]	Cuckoo	Windows	Ransomware 582	CryptoLocker	Resilient Information System Security (RISS) dataset	107
					Reveton		90
					Kovter		64
					Critroni		50
					TeslaCrypt		6
					Locker		97
					CryptoWall		46
					MATSNU		59
					KOLLAH		25
					GPCODER		4
					Trojan-Ransom		34
				Goodware			942
24	[88]	Cuckoo	Windows	Ransomware		VirusShare	1909
				Benign		Softonic	1139
25	[107]	Cuckoo Sandbox	Windows	Ransomware		VirusTotal	22,000
				Benign		Windows	100
26	[108]	Cuckoo	Edge Computing	Ransomware		Resilient Information Security System (RISS)	582
				Benign			942
27	[16]	Cuckoo Sandbox	Windows	Ransomware 1232	CryptoWall	VirusShare VirusTotal Malwarebytes Offensive-Computing	151
					Trojan Ransom		82
					TeslaCrypt		74
					Kollah		73
					Reveton		50
					Citroni		67
					TorrentLocker		108
					Pgpcoder		46
					Dirty Decrypt		51
					Kovter		23
					CryptoLocker		173
					Petya		89
					Cerber		171
					WannaCry		74
				Benign		Software-informer System 32 of Windows 7 Pro	1308
28	[109]	PIN tool and Custom Python program	Supervisory control and data acquisition systems (SCADA)	Ransomware		VirusTotal	561
				Benign		Windows	447
29	DeepGuard [110]	Cuckoo	Windows	Ransomware		VirusShare	2000
				Goodware		System logs	2000

Table 3. *Cont.*

Ransomware Datasets							
Sr#	**Study**	**Tool**	**Platform**	**Dataset Type**	**Family Name**	**Dataset Source**	**# of Samples**
30	[111]	Cuckoo Sandbox	Windows/Mac/Mobile	Ransomware		Sgandurra	582
				Benign			360
31	[112]	Did not mention	Desktop	Ransomware		VirusTotal	35,369
				Benign			43,191
32	DRDT [113]	Cuckoo	Windows	Ransomware		Sangfor Technologies Incorporation	1000
				Benign			1000
33	[114]	Cuckoo Sandbox	Windows	Ransomware		NA	80
				Non-Ransomware		NA	80
34	[115]	Genymotion	Mobile Android	Ransomware		VirusTotal	400
				Benign apps		GooglePlay Store	400
35	Peeler [116]	I/O patterns observation from Process execution patterns	Windows	Ransomware	Cerber	VirusTotal MalwareBazaar theZoo Malware samples from github	33
					Sodinokibi		14
					GoldenEye		12
					Sage		5
					Locky		5
					Dharma		3
					dotExe		3
					WannaCry		3
					Xorist		2
					Virlock.Gen.5		83
					LockScree.AGU		12
					Alphabet		2
					Other		29
				Benign			
36	[117]	Did not mention	Windows	Ransomware	CryptoWall	VirusTotal	17
					Deshacop		2
					CryptoDefense		6
					Upatre		56
					Zbot		6
					Critroni		2
					Yakes		150
					Crowti		23
					Others		10
				Benign		Windows 7, 8.1, 10 logs	NA
37	[118]	Cuckoo Sandbox	Windows	Ransomware 666	TeslaCrypt	VirusTotal	348
					CryptoShield		4
					Cerber		122
					Crysis		8
					Sage		5
					Unlock26		3
					Locky		129
					CryptoMix		2
					Petya		2
					WannaCry		1
					Flawed		1
				Benign		Software repository website	103

6. Research in Ransomware Detection

This study is classified according to the literature available in the field of ransomware detection. The available literature is categorized into two major categories, i.e., analysis and counteraction. Counteraction studies for the ransomware attacks were further categorized into prediction, prevention and detection [24].

6.1. Ransomware Analysis

Ransomware analysis plays a vital role in the detection of ransomware attacks. Ransomware analysis helps to understand the series of action performed by a program, figuring out its behavior and the types of operations associated with a malicious program. Ransomware analysis is classified into three categories, i.e., static, dynamic, and hybrid analysis [24]. Figure 5, gives the overview of the analysis studies discussed in this study. All the discussed analysis studies are summarized and presented in the Table 4.

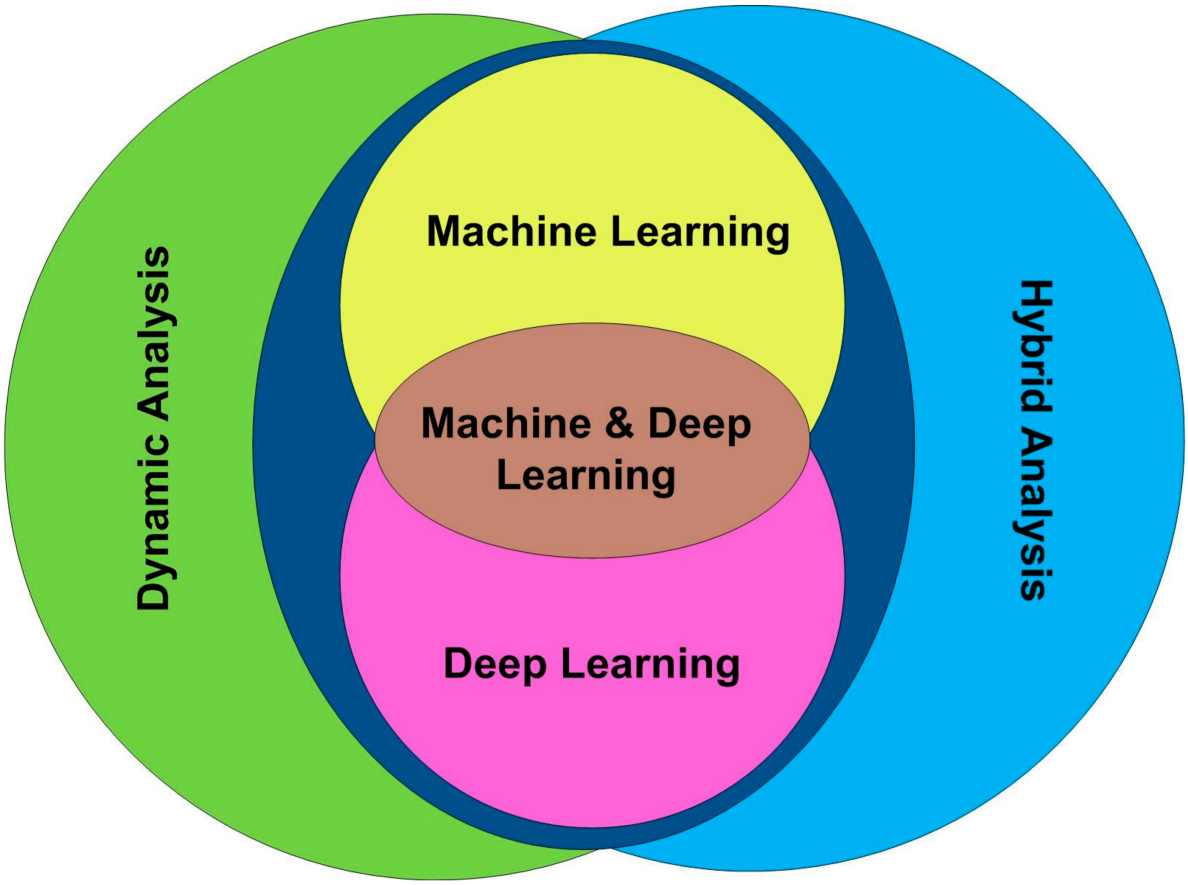

Figure 5. Ransomware analysis overview.

6.1.1. Static Analysis

Static analysis is a passive approach performed without executing the malicious code. It works by matching or comparing the features related to malicious code with already available features repository [24,27].

6.1.2. Dynamic Analysis

Dynamic analysis is an active approach performed by executing the malicious code. Malicious code run under controlled environment and exhibited features are captured by the controlled environment [24,27].

Table 4. Studies performed dynamic and hybrid analysis for ransomware detection.

Sr#	Study	Ransomware Analysis					
		Dynamic			Hybrid		
		ML	DL	ML and DL	ML	DL	ML and DL
1	[94]	✓					
2	[20]	✓					
3	[89]		✓				
4	[62]		✓				
5	[80]						✓
6	[85]		✓				
7	[90]		✓				
8	[81]		✓				
9	[119]	✓					
10	RANDS [96]	✓					
11	[97]		✓				
12	[7]	✓					
13	[98]	✓					
14	PEDA [79]				✓		
15	[99]	✓					
16	RanStop [100]		✓				
17	[101]	✓					
18	[91]			✓			
19	[95]	✓					
20	[102]	✓					
21	[118]	✓					
22	AIRaD [103]				✓		
23	[104]			✓			
24	[105]				✓		
25	[106]				✓		
26	[88]	✓					
27	[107]						✓
28	[108]		✓				
29	[16]		✓				
30	[109]		✓				
31	DeepGuard [110]		✓				
32	[112]	✓					
33	DRDT [113]		✓				

A system named EldeRan was presented in [58]. This system performed dynamic analysis to monitor the operations performed by the applications. API calls and strings were captured during runtime to observe the dynamic behavior of ransomware applications. Applications were monitored during their installation process to capture the ransomware discriminating features. This study was based on a concept that a ransomware sample exhibits its unique behavior in the actions performed in early phases of its installation.

A ransomware detection system consisting of 4 layer of deep belief network was proposed in [48]. This was an online detection solution for the embedded systems and IoT devices. System performed behavioral analysis. This detection solution utilized the optimized approach. Features were extracted from http requests and cryptographic system calls from the installation process of an application.

An automated ransomware detecting approach utilizing the LSTM, a deep learning technique was presented in [63]. To observe the behavior of a process, API calls were considered as point of focus. Cuckoo extracted the API calls using the API hooks. The study utilized the dynamic analysis to get the API calls sequence. However, this study run the analysis for 20 min which was the limitation of this study.

A detection system was proposed in [60] by combining sequential pattern mining for the feature identification with machine learning classification techniques. A dataset of 1624 ransomware samples were used to perform analysis. Bagging, J48, and RF were used as machine learning technique while multilayer perceptron was used as deep learning techniques. However, the samples were executed for only 10 s which can be easily evade by obfuscated ransomware samples. A greedy stepwise search method was used to find the distinctive features which identified the ransomware samples.

In [64], the proposed system applied the machine learning and deep learning technique multi-layer perceptron (MLP) for the ransomware detection and classification. The machine learning techniques included LR, NB, DT, RF, KNN and SVM algorithms. The effectiveness of shallow and deep networks was evaluated for the detection and classification of ransomwares. API invocations considered to help the evaluation. Experiments were trained using TensorFlow. Experiments were performed by using the cuckoo sandbox. Experimental configuration was performed on linear and radial kernel.

I. Ransomware Dynamic Analysis Studies Utilizing Machine Learning

Many ransomware detection studies used machine learning technologies to carry analysis. Some of the studies are discussed below.

A detection system in [94] performed an in-depth dynamic analysis to collect the system calls. This method selected the API calls that were most relevant in the detection process. This study introduced the refinement process to reduce the API calls from windows platform, hence enhancing the relevancy. Machine learning classifiers were trained on refined system call sequence data.

In [20], a system call based ransomware detection method was proposed. Dynamic analysis was performed to extract the related APIs from ransomware, malware, and benign samples. The features that met the fixed threshold were considered for the model training which played important role in the detection process.

An analysis based predictive android ransomware detection model was proposed in [119]. The study performed analysis by considering the available solutions for ransomware detection along with analysis on android apps to get the permissions dataset. The apps were scanned by using the weka platform. The predictive model was built by using machine learning classifiers.

A machine learning based multi-tier ransomware detection solution was presented in [96]. This detection system worked in three phases analysis, learning and detection. Behavioral analysis was performed to detect the unknown ransomware variants. However, the dynamic analysis was carried out for 5 min which made this system unsuitable for the advanced ransomware detection.

An automatic real-time ransomware detection system was developed in [7] which provided the protection to a hospital network. The system was integrated with Integrated

Clinical Environment (ICE). This system was able to detect and isolate the victim device to stop the spread of attack to other devices in the network. The system performed detection in 30 s. A network profile was built using ransomware traffic for half hour and the clean network traffic for six hours.

A study was conducted in [98] which utilized honeypots to collect the infected files from a host. Infected files were used to perform the dynamic analysis. This was a machine learning study aimed to predict and detect ransomware attacks. However, this study did not discuss details about carried out analysis.

An automated ransomware early detection and pattern extraction technique was presented in [99]. This system performed automatic dynamic analysis and extracted discriminating features. Real host logs from security operation centers were used to extract the features. This study did not mention the number of ransomware and normal samples, considered to perform analysis.

A crypto-ransomware detection system specifically focusing on locky ransomware was demonstrated in [101]. Network activities were analyzed using dedicated testbeds. Features were extracted from the network traffic observed at two different levels. Network packets and flow were monitored to capture PCAP files which later used to extract network features.

A ransomware detection method was proposed in [102] which utilized the process mining technique along with machine learning algorithms. Feature extraction from process mining was performed by using a fuzzy algorithm-based tool disco. Samples were run in the virtual machine and event logs were generated. A process monitoring tool was also used to monitor the registry activities and file system. However, details about analysis performed were not explained in the study.

A two-stage ransomware detection model was built in [89]. In the first stage the model collected windows API calls by performing the dynamic analysis in cuckoo sandbox. In second stage Markov chain model along with machine learning algorithm was utilized to classify the sample into benign or ransomware.

A model that extracted the runtime features and perform ransomware detection was proposed in [112]. Model performed detection during execution by scanning the API calls, registry activities, network, and file system. Behavior based features were extracted from API calls. Online machine learning classifiers were used to classify the samples.

A novel early detection system to detect the ransomware was presented in [95]. This detection system addressed the limitation of existing ibagging and bagging++. This system dealt with the limitation of little data availability in early stages of ransomware attacks by integration of ibagging and enhanced semi-random subspace selection techniques. Informative features subsets were built by incrementally dividing the features extracted from the dynamic analysis. The produced subsets were noise free and divergent with optimal selected features.

An upgraded ransomware detection model was presented in [118] which grouped the monitoring model and registry key operations. It was a behavioral detection model which run the dynamic analysis for maximum 60 min in cuckoo sandbox. It was a forward-looking monitoring system which utilized machine learning techniques for exploring and comparing the feature space. This system also provided the file backup on windows system.

II. Ransomware Dynamic Analysis Studies Utilizing Deep Learning

The ransomware detection studies which made the use of deep learning algorithms for the analysis and classification of ransomware are briefly summarized below:

A detection study named RAPPER was presented in [81] which worked in two modes, i.e., offline and online. This detection and prevention study carried out the dynamic analysis by using the sandbox. Ransomware and benign samples were observed for 10 ms to generate the profile for the normal processes. Datasets was generated by observing the system activities for 10 ms which was the limitation of the system. This detection system utilized LSTM, a deep learning autoencoder and fast Fourier transformation to detect the malicious activity.

In [62], a fog layer based ransomware detection system was presented. Dynamic analysis was performed on ransomware samples from different families. System utilized the deep learning techniques for the detection of unseen ransomware families. In this study the dynamic analysis was performed for only 10 s, making this system unsuitable for the detection of ransomware using evasion techniques.

A ransomware detection model for industrial IoT, utilizing the dynamic analysis method was proposed in [89]. Discriminating features were extracted from the API by performing the dynamic analysis. Two autoencoders named classical autoencoder and variational were used to detect the behavioral characteristics of the samples. A deep neural network classifier classified the program into ransomware or normal category.

A ransomware detector utilizing the deep learning was developed in [85] which was able to detect the ransomware events for bare metal server. The detector was named DeepRan that was cable of detecting the ransomware before it starts encryption in a network environment. Normal host logs and ransomware event logs were generated which then utilized to train the deep learning classifier. However, this study could not classify the zero-day ransomware attacks as the classifier was trained on 17 ransomware families.

A ransomware detection study in [90] utilized the variational autoencoder, an artificial neural network architecture to classify the fed sample in ransomware or benign. In the study it was mentioned the detection model was trained on the dynamically analyzed data from a set of 1524 samples of ransomware and benign. However, no details of analysis were presented.

A pre-encryption ransomware detection system was proposed in [97] which utilized the API for the detection purpose. The ransomware samples were analyzed in cuckoo to extract the API calls. This detection system first compared the under-examination sample with already available sample repository. In the second stage the samples were run in cuckoo to observe its behavior.

A hardware assisted ransomware detection technique was proposed in [100] named RanStop. This system was able to detect the ransomware in 2 ms before encryption happens. Hardware program counter (HPC), a monitoring micro architectural unit of modern processor was used to collect and dynamically analyzed the running events. However, it has limitation as the hardware data processed in runtime can also be corrupted.

A deep learning study analyzing the ransomware samples dynamically was proposed in [108]. This method was proposed for the edge computing devices to reduce the latency of data transmission. The behavioral analysis was carried out in cuckoo to extract the features for training. The deep learning classifier autoencoder was used to train on the extracted data and perform the detection.

An adaptive ransomware detection framework was proposed in [16]. This framework detected the ransomware by performing the dynamic analysis on a set of benign and ransomware. The dynamic analysis was performed for 4 to 9 min. The proposed deep learning semi supervised approach extracted the runtime behavioral patterns. Hence it was able to detect the zero-day ransomwares.

A scalable ransomware detection framework was proposed in [109] to detect the attack on electric vehicle charging stations (EVCS) controlled by supervisory control and data acquisition (SCADA) system. The dynamic binary analysis was performed in virtual unit via PIN framework. The proposed framework performed the analysis in two phases. Assembly frequency analysis was conducted offline on stored data in one phase while in online fashion using the real time data in another phase. Both features extraction and selection were performed by using the frequency analysis of assembly instructions.

A ransomware detection system named DeepGuard was proposed in [110] which detected the ransomware by observing the user file interacting pattern. A boundary was defined between legitimated user behavior and ransomware activities. Different ransomware families were executed in the sandbox to log the malicious activities and the file interaction patterns. The deep autoencoders were trained on the log data of user activities patterns.

A detection model [113] named Dynamic Ransomware Detector based on the improved TextCNN (DRDT) utilized the dynamic analysis to detect ransomware in TextCNN. The model was trained by using the API calls. This detection model improved the performance of TextCNN by using the API calls, NLP, and max pooling. Sandbox was used to process the API calls.

III. Ransomware Dynamic Analysis Studies Utilizing both Machine Learning and Deep Learning

There are some ransomware detection studies which performed the dynamic analysis while utilizing both learning technologies, i.e., machine and deep learning techniques to produce the optimal results.

A ransomware detection system for the backup systems, i.e., cloud services was introduced in [91]. Machine learning and deep learning were utilized to detect and classify the ransomware attack. Detection was performed based on optimal entropy reference value. Entropy of six different file formats were compared by using 600 normal files and 600 attack infected files. The system was also able to recover the loss due to compromised attack.

A behavioral dynamic analysis was performed in [104] to detect the high survivable ransomware (HSR). The dynamic analysis was performed in cuckoo for 4 to 9 min. The relevant features were extracted from the generated log files. The automated dynamic analysis was performed on the real-world ransomwares for windows platform.

6.1.3. Hybrid Analysis

Hybrid analysis is an analysis approach which takes the advantages of static and dynamic analysis to produce more accurate analysis result. In hybrid analysis, both static and dynamic features are used [77].

A mobile based ransomware detection model which utilized the advantages of static and dynamic analysis was presented in [65]. Machine learning classifiers NB, SVM, RF, and AdaBoost and DNN a deep learning algorithm, was used for the detection. The subjected app was checked statistically before submission to the dynamic analysis. If app was found suspicious then it was sent for the dynamic analysis. This system compared the malicious sample with ransomware DNA. By using the dynamic analysis, profile of each family was built.

A detection approach HELDROID was presented in [72], which performed detection by using the hybrid approach. The strings were analyzed statically after which the dynamic analysis was performed to check C&C traffic. Due to hybrid analysis this approach was able to detect unknown samples as well. Static analysis was used to make deterministic decisions.

A quantification model to prevent and detect the ransomware for the local drive was proposed in [120]. This model utilized hybrid analysis and social engineering for detection of ransomware. This model calculated the situational frequencies and possibility of registered and unregistered signatures. For social engineering characteristic and frequency analysis was performed to generate prediction.

I. Ransomware Hybrid Analysis Studies Utilizing Machine Learning

Many ransomware detection studies utilized the machine learning technologies to carry out the hybrid analysis. Some of the studies are explained below.

A pre-encryption ransomware detection system PEDA was proposed in [79]. This system was able to detect the ransomware before the encryption process starts. This study carried out hybrid analysis by using the signature comparison in first phase and analyzing the ransomware samples in second phase using cuckoo. However, samples were run for 30 s to capture the features from the subjected samples.

An AI based ransomware detection framework referred as AIRaD was proposed in [103]. This tool was able to detect ransomware by combining both static and dynamic analysis. A multi-level analysis was performed on assembly, DLL, and function calls. Dy-

namic analysis was performed with use of sandbox and PIN tool while reverse engineering was performed using Ghidra tools.

A multi-tier analytical model able to detect the ransomware attacks was proposed in [105]. It was a hybrid detection model utilized hybrid analysis as well as hybrid machine learning algorithms. Both static and dynamic traits were collected from semi realistic and realistic environment to detect the ransomware. Statistic and behavioral analysis were performed to detect the ransomware.

A pre-encryption ransomware detection system was proposed in [106], which utilized hybrid analysis approach. The system worked in two stages supporting two analyses. First stage performed static analysis supporting pre-execution while second stage performed dynamic analysis performing pre-encryption. This system also generated signature repository by dynamically analyzing the crypto ransomware.

II. Ransomware Hybrid Analysis Studies Utilizing Deep Learning

During the survey we did not find any study which utilized hybrid analysis along with deep learning methods.

III. Ransomware Hybrid Analysis Studies Utilizing Both Machine Learning and Deep Learning

Some of ransomware detection studies ripe off the advantages of both, machine learning and deep learning technologies to carry out more accurate analysis. Some of the studies are presented below.

A ransomware detection study performing feature analysis was presented in [80]. Two different datasets were generated to carry out static and dynamic analysis for the better detection. This study also utilized the machine and deep learning to perform the feature engineering. This study collected information from API calls and registry operations to make the decision. The study selected 300 dynamic features to calculate the accuracy.

A ransomware detection study [107] utilized both static and dynamic analysis to early detect the crypto-ransomware. To analyze the behavior, file content entropy and I/O activities were monitored to extract the features. The fast bare-metal sandbox system was used to capture the time series samples execution for five minutes.

6.1.4. Limitation of Analysis Performed in Ransomware Detection Studies

Ransomware detection studies carried out static, dynamic, and hybrid analysis to classify the program as ransomware or benign. These studies followed different set of parameters to run the analysis. However, there are still limitation while running the analysis for the fed samples. Some of the limitations are enlisted below:

1. Ransomware that are developed using its own encryption mechanism may evade analysis.
2. Static analysis has limited scope because of higher false alarms and limited accuracy.
3. Ransomware that fingerprints the environment, can bypass the analysis.
4. Few studies did not mention details about the dataset and analysis.
5. Analysis carried out for the fixed time could helped the evasion techniques.
6. Availability of limited amount of data in the initial stage of encryption process.
7. The ransomware using obfuscation and evasion techniques are difficult to discover.
8. Ransomware samples run for the short time could evade the detection.
9. Some of the studies lack in defining the source of dataset and number of samples used.
10. Some of the detection studies cannot detect the system API calls or encrypted operations that run during execution.
11. Runtime detection programs could be infected to malicious programs. The hardware data can be corrupted due to malicious program.

6.2. Ransomware Counteraction

Many studies are conducted to confront the ransomware attacks. These studies are further classified into three classes, i.e., detection, prevention, and prediction. Here we included specific type of studies while dealing with crypto-ransomware attacks [24,25].

6.2.1. Prediction

Prediction is aimed at stopping the ransomware attack before it takes place. It acts as a first step to stop occurrence of unfavorable outcomes. Information is gathered for analysis to make prediction about the possible attack [25].

An android based ransomware prediction and detection system was proposed in [119]. This permission-based system scanned all the apps in an uninfected mobile to extract all the permissions. The existing ransomware detection solutions were also reviewed to make a comprehensive set of app permissions. Data mining techniques were applied on extracted dataset to predict the occurrence of the attack. Machine learning classifiers were used to implement this system.

A ransomware prediction and detection system was presented in [98]. This system utilized honeypot concept to collect the data from trap files. A large set of 30,000 attributes from an online repository of machine learning, i.e., RISS of ICL were used to predict the occurrence of ransomware. For the feature selection five attributes were considered from large set of attributes. Six different machine learning algorithms were used to implement the detection system.

6.2.2. Prevention

Prevention studies aims at avoiding the occurrence of ransomware attack. It helps the potential user to be protected from being a victim to this attack. These studies are conducted with an aim to stop ransomware in the first place. It involves fixing the security holes in the system. Preventing the device from attack is easier than applying the remedy after occurrence of an attack. Prevention studies were further classified into Proactive and Reactive studies [24,25,38].

I. Proactive Prevention

In [55], a preventive technique was proposed for the android platform. It continuously monitored the processes and directories for ransomware detection. It utilized the statistical data collected from processor, memory, storage, and I/O devices to detect, and remove the ransomware. Process with abnormal behavior carried different statistics was stopped and terminated. This technique was faster as it implemented at kernel level in android source code. This technique was also able to detect ransomware with new patterns. User feedback was involved to decide the continuation of detected processes. Device performance was degraded after the use of proposed technique.

A ransomware detection model was presented in [66] which extracted payloads from real world network traffic. It was a preventive and detection model which could inspect the beginning of an attack. This detection technique utilized deep and machine learning to inspect the network traffic. Model executed in the static state and was trained on raw payloads and http requests.

An early ransomware detection and prevention model was developed in [83]. The prevention system perform functionality in 4 phases on unstructured data of Petya and Wannacry. Ransomware attack patterns and relevant features of ransomware attack were collected. Detection was performed by utilizing the machine learning algorithms.

In [93], a high survivable ransomware detection and prevention approach was presented. This approach prevent data from encryption by aborting the encrypting process. This approach stopped the encryption before it takes place. HSR used domain generation algorithm. This approach consisted of two modules, i.e., connection monitor and connection break. A process with no valid connection was mentioned suspicious. Once a connection was mentioned suspicious, all other users broke connection with it. This approach was based on stopping the key exchange to prevent the attack.

A proactive ransomware prevention model was proposed in [120]. This model also detected the ransomware attack by performing the hybrid analysis on black/whitelist and user list samples. This model utilized the social engineering to assess the occurrence

of a ransomware attack. The hybrid analysis and frequency analysis were performed to calculate the injury and conditional frequency analysis.

II. Reactive Prevention

A ransomware detection approach was proposed in [50] which also provided the data backup. This system was able to hold original data by using the garbage collector. This system performed detection by using the frequent read and rewrite requests on the storage devices. This system utilized the storage devices and its data, which is why it is called Self Defensible SSD.

A ransomware detection tool named RAPPER was presented in [81]. This tool performed ransomware detection in two steps. Running processes were observed by calculating the statistics from hardware performance counter (HPC). In the second step, analysis and detection was done. LSTM and Fast Fourier transformation were utilized to generate fast, reliable, and accurate result. This framework provided the backup if ransomware attack occurs.

A ransomware detection framework was proposed in [109] for Supervisory Control and Data Acquisition system (SCADA). The proposed framework assessed the impact of ransomware attack on SCADA. This framework worked in two phases to perform detection. This deep learning-based model worked on online and offline data. This system issued an automatic driven supervisory command to other stations if a ransomware attack occurs. This system also restored the data to cope with attack.

A ransomware detection system was proposed for the backup systems in [91]. This system was able to recover the data when the system was infected by the ransomware attack. Detection of a ransomware attack was performed by calculating the entropy of six different file formats. Optimal reference model and different machine and deep learning classifiers were used to detect and restore the original data.

Different ransomware defensive systems were proposed to detect and recover from the occurrence of the attack. In [54] a ransomware defense system was proposed which could detect and recover data with the use of SSD-insider++ system embedded in SSD. This system worked in two folds, i.e., detection and data recovery. Attack detection was tightly coupled with data recovery task. This study was limited in its implementation as assessing every I/O block, its header, and payload during runtime is infeasible. A study in [121] presented the current solution available to the ransomware attacks. The extraction of decryption key from the memory was also discussed in this work. Another ransomware prevention technique ransom protect algorithm was proposed in [122]. This ransomware prevention technique stopped the access to the files by locking them. No process could access the files without user permission hence avoided the occurrence of the ransomware attacks. A ransomware detection and recovery solution named Self-Recovery Service (SRS) for edge server of IoT network was proposed in [123]. This method was able to recover victims' files and the server services without any delay.

6.2.3. Detection

These studies are aimed at finding the ransomware attack during or after it takes place. There are two approaches for ransomware detection, i.e., structural, and behavioral [24,25].

A mobile ransomware detection study was proposed in [8]. This study detected the android ransomwares by using bytecodes and remove them from device. However, this study has limited scope as the samples were considered from top 10 ransomware families at that time. Secondly this approach lacks in disassembling all selected samples.

In [57], a lightweight an android ransomware detection system named R-PackDroid was proposed. This system utilized the supervised machine learning with system API calls to categorize the samples. The fed samples were labelled as ransomware, malware, or trusted application. It worked well for the detection of already discovered ransomware. However, this system was not accurate as it did not detect the code during execution. This system was not fully developed for android systems.

A deep learning based ransomware detection method was proposed in [48] which utilized Deep Belief Network. This detection system was designed for the embedded systems and IoT devices. This method worked by storing the bit streams in the memory. This system performed behavioral based detection by using the memory assisted stochastic dynamic fixed-point arithmetic.

A windows based detection system using dynamic analysis was presented in [14] named as UNVEIL. The system detected the ransomwares by observing the changes on the desktop using similarity score of desktop screenshots. It created the artificial environment using sandbox for the detection purpose and observed how ransomware interacted with environment. This system also discovered the previously unknown zero-day ransomwares. This system performed detection from live and real world fed data. This system worked by attaching to the file system drivers focusing on write and delete requests. This system run within kernel for the detection purpose. This system specifically detected file locker and screen locker ransomware.

A ransomware detection study for Cryptowall ransomware was presented in [76]. This detection system performed dynamic analysis by implementing the honeypots. The network traffic was observed to find out the malicious behavior. Once the malicious behavior found, the infected system was isolated.

A ransomware detection system was developed in [82] which worked in two modes, i.e., user mode and kernel modes. A normal profile was built by collecting the normal activities data from 50 workstations. However, this detection system did not ensure the complete protection of the user data from the occurrence of the ransomware attack.

In [93], a ransomware detection and prevention approach was presented for high survivable ransomware. User has right to abort the suspicious connection hence stop the encryption process. This model worked in two phases. One phase detected the connection and other phase brake the malicious connection on the user command.

A ransomware detection study was introduced in [61]. This detection technique used PSO and SVM to develop an evolutionary machine learning based approach. This system was developed for android platform that deals with imbalanced data. Feature selection of imbalanced data and optimization were point of focus of this approach.

Ransomware counteraction studies are summarized in Table 5.

A. Ransomware Detection Studies w.r.t. Pre and Post Encryption

In this study ransomware detection studies are categorized according to the available literature, i.e., two major categories, which are conventional and early detection. Conventional and early detection studies which utilized the machine and deep learning technologies for the ransomware detection purpose are considered in this paper. An overview of the detection studies with respect to encryption timeline and use of machine and deep learning is given in the Figure 6.

I. Conventional Detection Studies

In the convolutional studies whole data is observed to mark the program as ransomware or benign. The conventional detection models made the decisions by considering the entire runtime data. Some of these models fell in post-encryption phase while performing the detection [124,125].

i. Conventional Detection Studies Utilizing Machine Learning

Numerous conventional detection studies in the literature utilized the machine learning approach to detect the ransomware. Some of them are elaborated here after.

A non-signature based refinement method was developed in [94] to detect the ransomware. The API calls refinement process was introduced to reduce the size of the system calls by collecting the most relevant API calls using machine learning approach. Five machine learning classifiers were used including DT, KNN, LR, RF and SVM. Each sample was analyzed for more than one minute, i.e., 70 to 90 s, which made its implementation limited.

Table 5. Summary of ransomware counteraction studies.

Counteraction Studies					
Sr#	**Study**	**Detection**	**Prediction**	**Prevention**	
				Proactive	**Reactive**
1	[120]			✓	
2	[93]	✓		✓	
3	[66]			✓	
4	[119]		✓		
5	[76]	✓			
6	[8]	✓			
7	[55]			✓	
8	[14]	✓			
9	[98]		✓		
10	[50]				✓
11	[82]	✓			
12	[57]	✓			
13	[83]	✓		✓	
14	[48]	✓			
15	[61]	✓			
16	[81]				✓
17	[109]				✓
18	[91]				✓

A real-time API based machine learning ransomware detection model was proposed in [20]. A class frequency and non-class frequency method was used to extract the API calls which helped to detect the ransomware and malware accurately. Six different machine learning algorithms including RF, LR, NB, SGD, KNN, and SVM were employed in this study. The detection model was intended to perform a performance comparison of CF-NCF and TF-IDF.

A predictive ransomware detection study utilizing the machine learning was proposed in [119]. Study included two types of analysis. One analysis was carried out on available ransomware detection studies while other analysis was carried out on android ransomware and benign apps. These apps were scanned to collect the list of requested permission lists. Based on permission, the prediction was made.

A machine learning based detection solution was presented in [96] which utilized decision tree and naive bayes algorithms. First prediction about ransomware was done using decision tree while other prediction was reinforced by the naive bays decision. This system was able to detect the zero-day ransomware attacks. System has limited functionality as the analysis on the fed samples was done for only 5 min.

An integrated clinical environment real-time detection system was developed in [7] for the hospital. This system was able to detect and mitigate the ransomware attack effects using SDN. This system was able to detect and isolate the ransomware attacked device to stop the attack in the first place. Machine learning algorithms OCSVM, RF, NB were used to perform detection in 30 s.

An android ransomware detection study was presented in [126]. This study utilized DT, RF, AdaBoost, and gradient boosting machine learning algorithms. Two experiments were carried out on two different datasets differing in number of attributes considered. In one experiment five attributes were considered while in other ten attributes were considered.

Data splitting in training and testing datasets were performed by using the Multinominal NB, Gaussian NB, and Bernoulli NB.

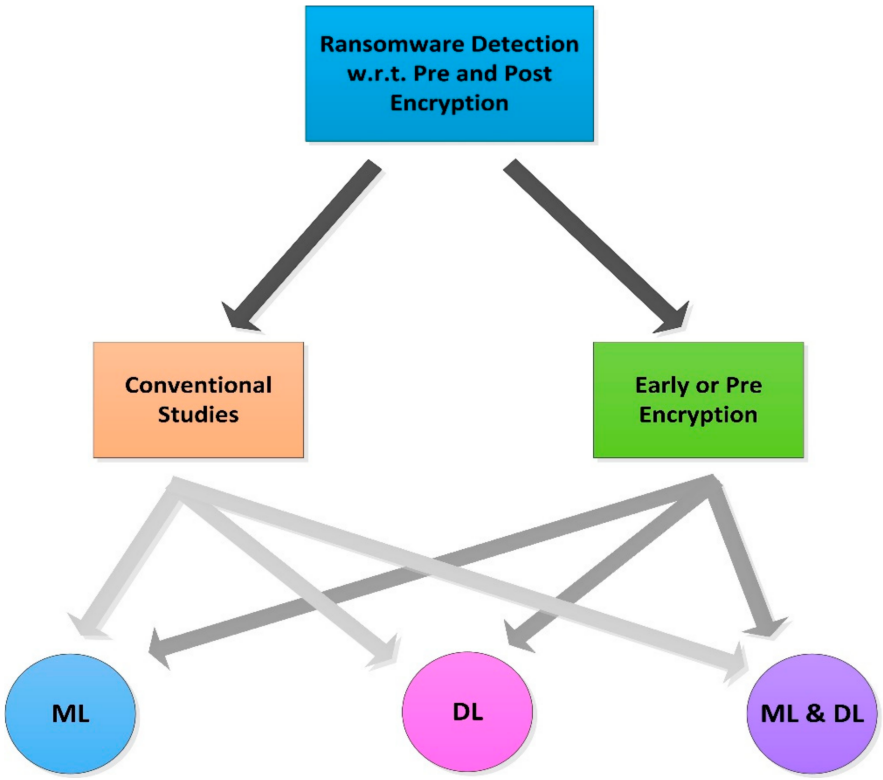

Figure 6. Overview of detection studies.

A supervised machine learning based prediction and detection study was explored in [98]. This study utilized the honeypots for detection of ransomware. A machine learning algorithm SVM was used to classify the ransomwares. The study did not highlight the dynamic analysis work properly.

A ransomware detection study was demonstrated in [101] which focused on analyzing the network activity. Locky ransomware was used to see its behavioral analysis and effects on the computer system. This study utilized four machine learning classifiers RF, LibSVM, random tree, and Bayes Net to perform the classification on two different levels, i.e., network packets and network flow. Network analyzing features were extracted from packets and flow of network.

In an android based ransomware detection study [32] the effectiveness of machine learning algorithms was evaluated. This study utilized a dataset CICAndMal2017, created by using the real android smartphones. Machine learning classifiers DT, RT, RF, NB, KNN, and SVM were trained and evaluated on two different dataset one containing 20% data from each family and second dataset contains whole ransomware dataset.

Process mining technique was used in [102] to detect the ransomware in 10 s. Event logs were utilized to find out number of occurrence of an event to detect the ransomware activity. Four different machine learning algorithms J48, NB, RF, and LR were used to perform classification. Two different tools disco and process monitoring were used to extract features from event logs and monitor the process for the ransomware detection respectively. However, study did not explain more details about the analysis and other aspects.

A system DNAact-Ran was proposed in [127] which utilized revolutionary digital DNA sequencing along with machine learning to detect the ransomware. This system was able to predict the ransomware due to the use of DNA sequencing. From a provided dataset, features were selected by using MOGWO and BCS and Digital DNA designed by

using k-mer frequency and DNA sequencing. The proposed system utilized NB, RF and SMO to classify the detected ransomware family.

An AI based ransomware detection framework AIRaD along with machine learning algorithms was proposed in [103]. Machine learning algorithms included in the work were SVM, LR, RF, and Adaboost with J48 and J48. Both behavioral and reverse engineering were used to detect the ransomware at multi-level. A signature repository was produced by using PIN tool, Cuckoo, and reverse engineering.

A multi-tier ransomware detection model was proposed in [105] which was able to detect the zero-day ransomware attacks. This hybrid detection model used static and dynamic traits to detect the ransomware from realistic and non-realistic environment. For realistic environment data was gathered for one month. This detection model classified the sample files into malware, ransomware, and benign. The infected events were observed both statistically and dynamically. Two machine learning algorithms DT and NB were used to classify the attack.

A machine learning based ransomware detection model was built in [88]. This model performed detection in two stages. Detection was performed by Markov model which was built by using the windows API calls. In the first stage dynamic analysis was performed in cuckoo along with the use of Markov model. In second stage machine learning algorithm random forest was used to perform detection

ii. Conventional Detection Studies Utilizing Deep Learning

Different conventional detection studies in the literature utilized the deep learning approach to detect the ransomware. Some of them are discussed below.

A detection and prevention tool named RAPPER was presented in [81] for the crypto-ransomware attacks. This study utilized the LSTM and Fast Fourier transformation to develop the detection system. Hardware performance counter was focused to detect the behavior of a malicious process, i.e., ransomware. The LSTM based autoencoders detected the unusual behavior of the process. The system activities were observed for 10 ms intervals which was the limitation of this system. After 10 ms the captured data was transformed into frequency domain by using the fast Fourier transformation. System provided backup if files encrypted from ransomware attack observed.

A fog layer ransomware detection system named DRTHIS was presented in [62]. System focused on the system call sequences to trace the ransomware attacks in 10 s. This detection system utilized two deep learning technologies, convolutional neural network, and long short-term memory for sample classification into ransomware or benign. This system was able to detect the zero-day ransomware families.

A ransomware detection study utilizing deep learning techniques was proposed in [89]. This detection model was proposed for the industrial IoT while dealing with unbalanced and high dimensional data. Two deep learning autoencoders were used to extract the features. One encoder performed unsupervised learning while other performed controlling operation. Both worked together to perform hybrid feature engineering. Data extracted after hybrid feature engineering was used to train the deep neural network.

A stacked variational autoencoder architecture was used to detect the ransomware using a blend of supervised and unsupervised learning methods [90]. This detection system was proposed to protect the industrial IoT systems. The study also utilized the data augmentation strategy to analyze data to increase the efficiency of the system. However, this study did not elaborate the analysis details.

A deep learning method was proposed in [108] to detect the ransomware attacks in edge computing devices. This method enabled the working of edge computing devices by protecting them from the ransomware attacks. Autoencoder of deep learning was used to train and classify the samples in benign and ransomware. The Autoencoder was trained on the behavioral features extracted by analyzing the samples in cuckoo sandbox.

An adaptive and scalable ransomware detection framework was proposed in [16] which was able to detect the evolving variants of ransomware. This was an anti-obfuscation framework utilizing the deep learning to perform detection. This framework combined

the unsupervised and semi-supervised model. However, this model did not mention its working in context of pre or post encryption phase.

A study in [109] was proposed to detect the ransomware attack on SCADA systems. Three deep learning algorithms CNN, DNN, and LSTM were utilized to detect the ransomware attacks. The system worked on stored and real-time data by implementing two phases, i.e., online, and offline phases respectively. Frequency analysis of assembly instructions was used for the features extraction and selection. This framework also provided the backup facility once a system was under attack.

DeepGuard a ransomware detection system was proposed in [110] which detected the ransomware from user activity behavior. The file interaction patters of user activity were used to define a profile. Behavior that deviated from the designed model was detected as ransomware. Three sigma limits were applied to generate the legitimated user behavior model along with use of deep generative autoencoders. The model was trained by using the dataset of legitimated user activities.

A Dynamic Ransomware Detector based on the improved TextCNN (DRDT) was proposed in [113], which detected the ransomware by utilizing TextCNN. The performance of TextCNN was improved by adding NLP and improving the pooling layer. Processes were monitored to record their behavior and registry modifications.

iii. Conventional Detection Studies Utilizing Both Machine Learning and Deep Learning

There are some conventional studies which implemented both machine learning and deep learning technologies for the ransomware detection purpose.

Ransomware detection study named RanSD technique was presented in [80]. This study performed feature engineering by utilizing the static and dynamic analysis along with utilization of machine and deep learning techniques. Mutual information was applied to find the distinct features. However, this study faced decrease in accuracy due to the presence of redundant features.

A file and behavioral based detection method was introduced in [91] for backup systems, i.e., cloud services. This system was able to detect the ransomware using entropy technique. Six different file formats were considered, and entropy of these file types were compared with normal and encrypted files entropies. This system was able to detect and recover the files due to compromised. This detection system utilized machine learning classifiers LR, KNN, RF, SVC, SVM, gradient boosting tree and deep learning classifier MLP to detect ransomware attack and recover the loss files.

A high survivable ransomware detection study was performed in [104], which utilized SVM and MLP algorithms to perform classification of ransomware. The proposed framework performed detection in three phases. The features were selected by using term frequency inverse document frequency. The performance comparison was performed with respect to other anti-viruses and classifiers.

II. Early Detection or Pre-Encryption Studies

In early detection or pre-encryption detection systems, the attack is detected before the encryption process starts. Here detection is invoked using the subset of runtime data. The early detection system also acts as a prevention technique to stop the encryption in first place [125,128]. Following are the studies that employed the pre-encryption detection concept.

i. Early Detection Studies Utilizing Machine Learning

Different early detection studies in the literature utilized the machine learning approach to detect the ransomware. Some of them are discussed here after.

A crypto-ransomware detection system was proposed in [97] which can detect the ransomware attack before encryption process started. This detection system utilized machine learning algorithm RF for decision making. Detection was performed in two stages by utilizing a combination of signature and behavioral detection. This detection system was evaluated using six evaluation metrics which was distinction of this work.

An early ransomware detection system PEDA was proposed in [79] which was able to detect the known and zero-day ransomware attacks before encryption started. This study utilized the signature and behavioral approach to detect the ransomware working in two stages. This study used RF algorithm to perform classification on the fed samples. This system was having many limitations including dependence on the windows APIs and was able to detect only one type of ransomware. This system was not able to detect the ransomware using its native code. This system was not able to fully detect the ransomware attacks instead it was proposed as a supplement detection system.

An automated early detection system was presented in [99] which performed testing on machine learning approaches using the pattern extraction. Seven ransomware attacks were considered in the study to test the performance of machine learning approaches. This tool was able to detect the ransomware before encryption starts by using the real logs from Security Operation Centers (SOC).

A pre-encryption ransomware detection and prevention system named was proposed in [106]. This system worked in two phases. One phase performed pre-execution by matching the signatures while second phase performed pre-encryption by dynamically analyzing the samples in cuckoo sandbox. This detection system was able to detect known and unknown ransomware before the encryption started. However, it was proposed that the system will have a high false positive rate.

ii. Early Detection Studies Utilizing Deep Learning

Some of the studies in ransomware detection literature, utilized the deep learning methods to detect the ransomware. Some of the studies are described below.

A deep learning-based ransomware early detection system named DeepRan was developed in [85] for bare metal server. The detector was able to detect ransomware before it started propagation. This system worked on anomaly detection technique. The normal profile was generated by using the normal activities on the host computer. A comparison was also performed for the developed system and commonly used deep learning models. However, the system was limited in terms of classification as it was trained on logs generated by 17 ransomware families.

A ransomware detection system RanStop was proposed in [100] which utilized the hardware assistance to detect the ransomware in 2 ms. This system was able to detect the known and unknown ransomware before the encryption starts. RNN and LSTM were used to train the system to detect the ransomware in runtime fashion. However, this system was limited in term of its functionality and implementation, i.e., required access to sufficient hardware resources.

iii. Early Detection Studies Utilizing Both Machine Learning and Deep Learning

The ransomware detection study that utilized the machine and deep learning algorithms to assist the detection is mentioned below.

A hybrid analysis-based ransomware early detection study was presented in [107]. Based on behavior, the machine learning classifier SVM and deep learning classifier LSTM were used to detect the ransomware. File content entropy and I/O activities were observed to extract the features. Data augmentation keyed method was used to synthesize the time series samples for early detection. The samples were executed in bare-metal sandbox for five minutes.

B. Limitations of Ransomware Detection Studies

Many studies have been conducted on the topic of ransomware detection. There are still limitations attached to each work. The limitations of existing detection work are listed below:

1. Most of the conducted studies for the ransomware detection fall under conventional class where ransomware is detected after the encryption starts.
2. In the literature there is not a single study that deals with population drift concept while considering the pre-encryption early detection.

3. High number of irrelevant and redundant system calls used to bypass the detection.
4. Developed ransomware studies used different number of logs from different ransomware families.
5. The ransomware detection systems are platform dependent. A system developed for windows API cannot be implemented for cloud and mobile devices.
6. Ransomware detection study cannot detect the ransomware which encrypt data using its own native code.
7. Not all the detection studies available in the literature are practical to implement. Some of the presented studies are empirical or supplement detection systems.
8. Honeypot methods are not fully reliable as there is no guarantee the honeypot folders will always be accessed by the attack.
9. Analyzing the samples for limited or ample time made the detection studies inadequate to implement.
10. Dealing with little amount of data or massive data with high redundant values.
11. Some of the studies did not explain well about the analysis performed for the detection.
12. There are few studies that detected ransomware for the backup systems. More studies are needed in this domain.
13. Datasets used to train data are synthetic and are extracted from specific sources, i.e., pseudo real world events.

A summary of conventional and early ransomware detection is given in Table 6.

Table 6. Summary of Ransomware studies w.r.t Pre and Post Encryption.

Detection Studies w.r.t. Pre and Post Encryption							
Sr#	**Study**	**Conventional**			**Early/Pre-Encryption**		
		ML	**DL**	**ML and DL**	**ML**	**DL**	**ML and DL**
1	[94]	✓					
2	[20]	✓					
3	[89]		✓				
4	DRTHIS [62]		✓				
5	RanSD [80]			✓			
6	DeepRan [85]					✓	
7	[90]		✓				
8	RAPPER [81]		✓				
9	[119]	✓					
10	RANDS [96]	✓					
11	[97]				✓		
12	[7]	✓					
13	[126]	✓					
14	[98]	✓					
15	PEDA [79]				✓		

Table 6. *Cont.*

Detection Studies w.r.t. Pre and Post Encryption							
Sr#	**Study**	**Conventional**			**Early/Pre-Encryption**		
		ML	**DL**	**ML and DL**	**ML**	**DL**	**ML and DL**
16	[99]				✓		
17	RanStop [100]					✓	
18	[101]	✓					
19	[91]			✓			
20	[32]	✓					
21	[102]	✓					
22	[127]	✓					
23	[103]	✓					
24	[104]			✓			
25	[105]	✓					
26	[106]				✓		
27	[88]	✓					
28	[107]						✓
29	[108]		✓				
30	[16]		✓				
31	[109]		✓				
32	DeepGuard [110]		✓				
33	DRDT [113]		✓				

7. Research Direction

In this paper a brief survey about the machine and deep learning technologies used in the field of ransomware detection was presented. After surveying the available literature, few research directions are highlighted below as there is still room of improvement in the field of ransomware detection. Several open issues that need further research are listed to improve the efficiency of the ransomware detection systems.

1. Dealing with computational and time complexities: The detection systems should be developed considering the computational overhead. A system should detect the ransomware attack in no time to make it valid. There should be detection systems with less time complexities. Specially for the devices with resources constrains, i.e., IoT and embedded systems.
2. Dealing with hardware complexities: Most of the developed systems are hard disk supported having high RAM. System with limited hardware should also considered while developing a detection system. The more complex detection systems will incur the high cost. Some of the available solutions are also hardware dependent requiring advance hardware.
3. Evasion and obfuscation: Ransomware development and detection is a non-stationary field which keep on evolving with time. Developing ransomware detection solutions should cope with ransomware evasion and obfuscation techniques. System which can deal with evasion and obfuscation will be more reliable with high accuracy and low false alarms.

4. Real-time system: Most of the available studies in the ransomware detection are empirical and proposed. A real-time detection system with minimal time to response to the attack is required. Failing to provide the real-time detection will lead to irreversible encryption of some or all the data.
5. Distributed environment: Most of the available ransomware detection solutions are developed for the desktops and mobile devices. There should more research on the spread of ransomware attack in a distributed environment. In a distributed network environment, most damage can be caused just by attacking a single device.
6. Scalable and adaptive: The developed solutions for the ransomware should be scalable so that they can deal with detection on multiple machines, i.e., improving the detection capabilities. The new ransomware detection model should be scalable to the advance ransomware and newly updated datasets. With the passage of time new and more sophisticated ransomware samples are being developed. These advanced ransomwares can be dealt well with the adaptive and scalable ransomware detection model.
7. Forensic by design concept: The integration of forensics by design and ransomware detection system would reduce the loss. Use of forensic by design will help to detect, mitigate, and roll back the ransomware attacks. This will also help in forensic investigation.
8. Use of fuzzy modelling: With the evolution of ransomware attacks there should be an automatic computation system for the ransomware. This will assess the damage to a zero-day ransomware attack. This fuzzy system will also help to detect the ransomware attacks based on the feature obtained after automatic analysis.
9. Use of stream data mining: Stream data mining can be used to reduce the ransomware detection time as this method works well with high dimensional and rapidly changing data. The integration of stream data mining will also bring the optimal solution for ransomware detection.
10. Feature reduction for deep learning models: The deep learning models requires plenty of data to get train and make decision. While in the ransomware detection only little data is available before encryption happens. So, there should be feature reduction methods for the deep learning networks.
11. Development of Pre-Encryption detection systems: There are few studies available that are providing solution to early detection of ransomware attacks but still limited in the scope. Early detection is vital to restrain the ransomware attacks. Available solutions are limited and do not fit well to the advanced variants of ransomware.
12. Rich Dataset: There should be a dataset that contains all the ransomware attack patterns which could be used to train the machine and deep learning models. This model should be updated periodically. There is not a single dataset available that can be used as a benchmark to the developing ransomware detection systems.
13. Population drift: Ransomware is being populated day by day. Its variants are becoming more sophisticated with time. There should be detection studies which deal with population drift concept of ransomware.

8. Conclusions

In this paper, we presented a comprehensive survey of ransomware, ransomware related concepts, and ransomware detection approaches utilizing machine and deep learning technologies. A brief survey of ransomware detection techniques for different platforms was also presented. Datasets utilized with different machine and deep learning methods in the ransomware detection studies were also enlisted in this work. We provided an extensive overview about the dynamic analysis carried out in ransomware attacks detection. We proposed a taxonomy along with related concepts of ransomware detection studies. Ransomware detection studies which utilized the machine learning, deep learning and a blend of both technologies simultaneously were examined while categorizing them in the traditional studies, i.e., conventional, and early detection. Ransomware detection studies were also classified with respect to victim's platform. It represented the prevalence of attacks on different platforms. It highlighted the platforms that encountered more attacks. A list of research directions which need to be

focused for more research was also presented. A strong and accurate ransomware detection system can be obtained by a synergic relation of machine and deep learning technologies. This survey is intended to provide a user manual that can encourage researchers as a direction to work with available technologies in the field of ransomware attack detection. It can help them in developing the more efficient ransomware detection models while considering the available solutions. In the future, we shall work on the significance and contribution of static analysis for the detection of ransomware attacks utilizing machine and deep learning methods.

Author Contributions: Conceptualization, U.U. and B.A.S.A.-r.; methodology, B.A.S.A.-r. and A.Z.; validation, B.A.S.A.-r., F.A.G. and M.A.R.; formal analysis, U.U.; investigation, U.U. and B.A.S.A.-r.; resources, M.A.R.; data curation, U.U. and A.Z.; writing—U.U. and B.A.S.A.-r.; writing—review and editing, B.A.S.A.-r. and U.U.; visualization, U.U.; supervision, B.A.S.A.-r., A.Z. and F.A.G.; project administration, M.A.R.; funding acquisition, B.A.S.A.-r. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: There is no conflicts of interest.

Nomenclature

The following acronyms are used in this manuscript.

Acronym	Description
TP	True Positive
NB	Naive Bayes
BN	Bayes Network
LR	Logistic Regression
RF	Random Forest
RT	Random Tree
DT	Decision Tree
ML	Machine Learning
DL	Deep Learning
IoT	Internet of Things
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
SVC	Support Vector Classifier
SDN	Software Defined Networking
SGD	Stochastic Gradient Descent
SSD	Solid State Drive
DLL	Dynamic Link Library
CPU	Central Processing Unit
RNN	Recurrent Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
LMT	Logistic Model Tree
MLP	Multilayer Perceptron
NLP	Natural Language Processing
PSO	Particle Swarm Optimization
BCS	Binary Cuckoo Search
GIGO	Garbage In, Garbage Out
LSTM	Long Short-Term Memory
MOGWO	Multi-Objective Grey Wolf Optimization
LIBSVM	Library for Support Vector Machine

References

1. Khalaf, B.A.; Mostafa, S.A.; Mustapha, A.; Mohammed, M.A.; Mahmoud, M.A.; Al-Rimy, B.A.S.; Abd Razak, S.; Elhoseny, M.; Marks, A. An Adaptive Protection of Flooding Attacks Model for Complex Network Environments. *Secur. Commun. Netw.* **2021**, *2021*, 5542919. [CrossRef]
2. Maseer, Z.K.; Yusof, R.; Mostafa, S.A.; Bahaman, N.; Musa, O.; Al-rimy, B.A.S. DeepIoT. IDS: Hybrid Deep Learning for Enhancing IoT Network Intrusion Detection. *CMC Comput. Mater. Contin.* **2021**, *69*, 3945–3966.
3. Young, A.; Yung, M. Cryptovirology: Extortion-Based security threats and countermeasures. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 6–8 May 1996; pp. 129–140.
4. Aboaoja, F.A.; Zainal, A.; Ghaleb, F.A.; Al-rimy, B.A.S. Toward an Ensemble Behavioral-Based Early Evasive Malware Detection Framework. In Proceedings of the 2021 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 6–7 October 2021; pp. 181–186.
5. Alghofaili, Y.; Albattah, A.; Alrajeh, N.; Rassam, M.A.; Al-rimy, B.A.S. Secure Cloud Infrastructure: A Survey on Issues, Current Solutions, and Open Challenges. *Appl. Sci.* **2021**, *11*, 9005. [CrossRef]
6. Zavarsky, P.; Lindskog, D. Experimental analysis of ransomware on windows and android platforms: Evolution and characterization. *Procedia Comput. Sci.* **2016**, *94*, 465–472.
7. Fernandez Maimo, L.; Huertas Celdran, A.; Perales Gomez, A.L.; Garcia Clemente, F.J.; Weimer, J.; Lee, I. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. *Sensors* **2019**, *19*, 1114. [CrossRef]
8. Mercaldo, F.; Nardone, V.; Santone, A.; Visaggio, C.A. Ransomware steals your phone. Formal methods rescue it. In Proceedings of the International Conference on Formal Techniques for Distributed Objects, Components, and Systems, Heraklion, Crete, 6–9 June 2016; Springer: Cham, Switzerland, 2016; pp. 212–221.
9. Scaife, N.; Carter, H.; Traynor, P.; Butler, K.R. Cryptolock (and drop it): Stopping ransomware attacks on user data. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016; pp. 303–312.
10. Forestiero, A.; Mastroianni, C.; Spezzano, G. A Multi-Agent Approach for the. *Self-Organ. Auton. Inform.* **2005**, *135*, 220.
11. Comito, C.; Forestiero, A.; Pizzuti, C. Word embedding based clustering to detect topics in social media. In Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki, Greece, 14–17 October 2019; pp. 192–199.
12. Forestiero, A.; Mastroianni, C.; Meo, M.; Papuzzo, G.; Sheikhalishahi, M. Hierarchical approach for green workload management in distributed data centers. In *European Conference on Parallel Processing*; Springer: Cham, Switzerland, 2014; pp. 323–334.
13. Kharraz, A.; Robertson, W.; Balzarotti, D.; Bilge, L.; Kirda, E. Cutting the gordian knot: A look under the hood of ransomware attacks. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Milan, Italy, 9–10 July 2015; Springer: Cham, Switzerland, 2015; pp. 3–24.
14. Kharaz, A.; Arshad, S.; Mulliner, C.; Robertson, W.; Kirda, E. {UNVEIL}: A large-scale, automated approach to detecting ransomware. In Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security 16), Austin, TX, USA, 10–12 August 2016; pp. 757–772.
15. Popli, N.K.; Girdhar, A. Behavioural analysis of recent ransomwares and prediction of future attacks by polymorphic and metamorphic ransomware. In *Computational Intelligence: Theories, Applications and Future Directions—Volume II*; Springer: Singapore, 2019; pp. 65–80.
16. Sharmeen, S.; Ahmed, Y.A.; Huda, S.; Koçer, B.Ş.; Hassan, M.M. Avoiding future digital extortion through robust protection against ransomware threats using deep learning based adaptive approaches. *IEEE Access* **2020**, *8*, 24522–24534. [CrossRef]
17. Al-Rimy, B.A.S.; Maarof, M.A.; Alazab, M.; Alsolami, F.; Shaid, S.Z.M.; Ghaleb, F.A.; Al-Hadhrami, T.; Ali, A.M. A pseudo feedback-based annotated TF-IDF technique for dynamic crypto-ransomware pre-encryption boundary delineation and features extraction. *IEEE Access* **2020**, *8*, 140586–140598. [CrossRef]
18. A Ghaleb, F.; Saeed, F.; Al-Sarem, M.; Ali Saleh Al-rimy, B.; Boulila, W.; Eljialy, A.; Aloufi, K.; Alazab, M. Misbehavior-Aware on-demand collaborative intrusion detection system using distributed ensemble learning for VANET. *Electronics* **2020**, *9*, 1411. [CrossRef]
19. Geluvaraj, B.; Satwik, P.; Kumar, T.A. The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace. In Proceedings of the International Conference on Computer Networks and Communication Technologies, Coimbatore, India, 23–24 May 2019; Springer: Singapore, 2019; pp. 739–747.
20. Bae, S.I.; Lee, G.B.; Im, E.G. Ransomware detection using machine learning algorithms. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5422. [CrossRef]
21. Krishnan, K.S.; Thampi, S.M. Deep Learning Approaches for IoT Security in the Big Data Era. In *Combating Security Challenges in the Age of Big Data*; Springer: Cham, Switzerland, 2020; pp. 105–135.
22. Faris, H.; Habib, M.; Almomani, I.; Eshtay, M.; Aljarah, I. Optimizing extreme learning machines using chains of salps for efficient Android ransomware detection. *Appl. Sci.* **2020**, *10*, 3706. [CrossRef]
23. Al-rimy, B.A.S.; Maarof, M.A.; Prasetyo, Y.A.; Shaid, S.Z.M.; Ariffin, A.F.M. Zero-day aware decision fusion-based model for crypto-ransomware early detection. *Int. J. Integr. Eng.* **2018**, *10*, 82–88. [CrossRef]
24. Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Comput. Secur.* **2018**, *74*, 144–166. [CrossRef]

25. Herrera Silva, J.A.; Barona López, L.I.; Valdivieso Caraguay, Á.L.; Hernández-Álvarez, M. A Survey on Situational Awareness of Ransomware Attacks—Detection and Prevention Parameters. *Remote Sens.* **2019**, *11*, 1168. [CrossRef]
26. Aurangzeb, S.; Aleem, M.; Iqbal, M.A.; Islam, M.A. Ransomware: A Survey and Trends. *J. Inf. Assur. Secur.* **2017**, *6*, 48–58.
27. Kok, S.; Abdullah, A.; Jhanjhi, N.; Supramaniam, M. Ransomware, threat and detection techniques: A review. *Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 136.
28. Alzahrani, N.; Alghazzawi, D. A Review on Android Ransomware Detection Using Deep Learning Techniques. In Proceedings of the 11th International Conference on Management of Digital EcoSystems, Limassol, Cyprus, 12–14 November 2019; pp. 330–335.
29. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.; Du, X.; Ali, I.; Guizani, M. A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
30. Tariq, M.I.; Memon, N.A.; Ahmed, S.; Tayyaba, S.; Mushtaq, M.T.; Mian, N.A.; Imran, M.; Ashraf, M.W. A Review of Deep Learning Security and Privacy Defensive Techniques. *Mob. Inf. Syst.* **2020**, *2020*, 6535834. [CrossRef]
31. Berman, D.S.; Buczak, A.L.; Chavis, J.S.; Corbett, C.L. A survey of deep learning methods for cyber security. *Information* **2019**, *10*, 122. [CrossRef]
32. Noorbehbahani, F.; Rasouli, F.; Saberi, M. Analysis of machine learning techniques for ransomware detection. In Proceedings of the 2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), Mashhad, Iran, 28–29 August 2019; pp. 128–133.
33. Fernando, D.W.; Komninos, N.; Chen, T. A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques. *IOT* **2020**, *1*, 30. [CrossRef]
34. Oz, H.; Aris, A.; Levi, A.; Uluagac, A.S. A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions. *arXiv* **2021**, arXiv:2102.06249.
35. Sharma, S.; Kumar, R.; Rama Krishna, C. A survey on analysis and detection of Android ransomware. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6272. [CrossRef]
36. Kumari, M. Application of Machine Learning and Deep Learning in Cybercrime Prevention—A Study. *Int. J. Trend Res. Dev.* **2019**, 1–4.
37. KP, S. A short review on Applications of Deep learning for Cyber security. *arXiv* **2018**, arXiv:1812.06292.
38. Humayun, M.; Jhanjhi, N.; Alsayat, A.; Ponnusamy, V. Internet of things and ransomware: Evolution, mitigation and prevention. *Egypt. Inform. J.* **2020**, *22*, 105–117. [CrossRef]
39. Reshmi, T. Information security breaches due to ransomware attacks-a systematic literature review. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100013. [CrossRef]
40. Olaimat, M.N.; Maarof, M.A.; Al-rimy, B.A.S. Ransomware Anti-Analysis and Evasion Techniques: A Survey and Research Directions. In Proceedings of the 2021 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia, 29–31 January 2021; pp. 1–6.
41. Hu, J.W.; Zhang, Y.; Cui, Y.P. Research on Android ransomware protection technology. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2020; p. 012004.
42. Maigida, A.M.; Olalere, M.; Alhassan, J.K.; Chiroma, H.; Dada, E.G. Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms. *J. Reliab. Intell. Environ.* **2019**, *5*, 67–89. [CrossRef]
43. Sharma, B.; Mangrulkar, R. Deep learning applications in cyber security: A comprehensive review, challenges and prospects. *Int. J. Eng. Appl. Sci. Technol.* **2019**, *4*, 148–159. [CrossRef]
44. Bello, I.; Chiroma, H.; Abdullahi, U.A.; Gital, A.Y.u.; Jauro, F.; Khan, A.; Okesola, J.O.; Shafi'i, M.A. Detecting ransomware attacks using intelligent algorithms: Recent development and next direction from deep learning and big data perspectives. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 8699–8717. [CrossRef]
45. Sneha, M.; Arya, A.; Agarwal, P. Ransomware Detection techniques in the Dawn of Artificial Intelligence: A Survey. In Proceedings of the 2020 the 9th International Conference on Networks, Communication and Computing, Tokyo, Japan, 18–20 December 2020; pp. 26–33.
46. Urooj, U.; Maarof, M.A.B.; Al-rimy, B.A.S. A proposed Adaptive Pre-Encryption Crypto-Ransomware Early Detection Model. In Proceedings of the 2021 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia, 29–31 January 2021; pp. 1–6.
47. Mbol, F.; Robert, J.-M.; Sadighian, A. An efficient approach to detect torrentlocker ransomware in computer systems. In Proceedings of the International Conference on Cryptology and Network Security, Milan, Italy, 14–16 November 2016; Springer: Cham, Switzerland, 2016; pp. 532–541.
48. Alrawashdeh, K.; Purdy, C. Ransomware detection using limited precision deep learning structure in fpga. In Proceedings of the NAECON 2018-IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 23–26 July 2018; pp. 152–157.
49. Feng, Y.; Liu, C.; Liu, B. Poster: A new approach to detecting ransomware with deception. In Proceedings of the 38th IEEE Symposium on Security and Privacy Workshops, San Jose, CA, USA, 22–24 May 2017.
50. Paik, J.-Y.; Shin, K.; Cho, E.-S. Poster: Self-defensible storage devices based on flash memory against ransomware. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 23–25 May 2016.
51. Bhardwaj, A.; Avasthi, V.; Sastry, H.; Subrahmanyam, G. Ransomware digital extortion: A rising new age threat. *Indian J. Sci. Technol.* **2016**, *9*, 1–5. [CrossRef]

52. Al-Rimy, B.A.S.; Maarof, M.A.; Alazab, M.; Shaid, S.Z.M.; Ghaleb, F.A.; Almalawi, A.; Ali, A.M.; Al-Hadhrami, T. Redundancy coefficient gradual up-weighting-based mutual information feature selection technique for crypto-ransomware early detection. *Future Gener. Comput. Syst.* **2021**, *115*, 641–658. [CrossRef]
53. Gazet, A. Comparative analysis of various ransomware virii. *J. Comput. Virol.* **2010**, *6*, 77–90. [CrossRef]
54. Baek, S.; Jung, Y.; Mohaisen, A.; Lee, S.; Nyang, D. SSD-Assisted Ransomware Detection and Data Recovery Techniques. *IEEE Trans. Comput.* **2020**, *70*, 1762–1776. [CrossRef]
55. Song, S.; Kim, B.; Lee, S. The effective ransomware prevention technique using process monitoring on android platform. *Mob. Inf. Syst.* **2016**, *2016*, 2946735. [CrossRef]
56. Gómez-Hernández, J.A.; Álvarez-González, L.; García-Teodoro, P. R-Locker: Thwarting ransomware action through a honeyfile-based approach. *Comput. Secur.* **2018**, *73*, 389–398. [CrossRef]
57. Maiorca, D.; Mercaldo, F.; Giacinto, G.; Visaggio, C.A.; Martinelli, F. R-PackDroid: API package-based characterization and detection of mobile ransomware. In Proceedings of the Symposium on Applied Computing, Marrakech, Morocco, 3–7 April 2017; pp. 1718–1723.
58. Sgandurra, D.; Muñoz-González, L.; Mohsen, R.; Lupu, E.C. Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. *arXiv* **2016**, arXiv:1609.03020.
59. Alhawi, O.M.; Baldwin, J.; Dehghantanha, A. Leveraging machine learning techniques for windows ransomware network traffic detection. In *Cyber Threat Intelligence*; Springer: Cham, Switzerland, 2018; pp. 93–106.
60. Homayoun, S.; Dehghantanha, A.; Ahmadzadeh, M.; Hashemi, S.; Khayami, R. Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence. *IEEE Trans. Emerg. Top. Comput.* **2017**, *8*, 341–351. [CrossRef]
61. Almomani, I.; Qaddoura, R.; Habib, M.; Alsoghyer, S.; Al Khayer, A.; Aljarah, I.; Faris, H. Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data. *IEEE Access* **2021**, *9*, 57674–57691. [CrossRef]
62. Homayoun, S.; Dehghantanha, A.; Ahmadzadeh, M.; Hashemi, S.; Khayami, R.; Choo, K.-K.R.; Newton, D.E. DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer. *Future Gener. Comput. Syst.* **2019**, *90*, 94–104. [CrossRef]
63. Maniath, S.; Ashok, A.; Poornachandran, P.; Sujadevi, V.; Sankar, A.P.; Jan, S. Deep learning LSTM based ransomware detection. In Proceedings of the 2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE), Noida, India, 26–27 October 2017; pp. 442–446.
64. Vinayakumar, R.; Soman, K.; Velan, K.S.; Ganorkar, S. Evaluating shallow and deep networks for ransomware detection and classification. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 259–265.
65. Gharib, A.; Ghorbani, A. Dna-droid: A real-time android ransomware detection framework. In *International Conference on Network and System Security*; Springer: Cham, Switzerland, 2017; pp. 184–198.
66. Tseng, A.; Chen, Y.; Kao, Y.; Lin, T. Deep learning for ransomware detection. *IEICE Tech. Rep.* **2016**, *116*, 87–92.
67. Kianpour, M.; Wen, S.-F. Timing attacks on machine learning: State of the art. In Proceedings of the SAI Intelligent Systems Conference, London, UK, 5–6 September 2019; pp. 111–125.
68. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. *arXiv* **2016**, arXiv:1611.01236.
69. Goodfellow, I.; McDaniel, P.; Papernot, N. Making machine learning robust against adversarial inputs. *Commun. ACM* **2018**, *61*, 56–66. [CrossRef]
70. Ameer, M. Android Ransomware Detection Using Machine Learning Techniques to Mitigate Adversarial Evasion Attacks. Ph.D. Thesis, Capital University of Science and Technology, Islamabad, Pakistan, 2019.
71. Cara, F.; Scalas, M.; Giacinto, G.; Maiorca, D. On the Feasibility of Adversarial Sample Creation Using the Android System API. *Information* **2020**, *11*, 433. [CrossRef]
72. Andronio, N.; Zanero, S.; Maggi, F. Heldroid: Dissecting and detecting mobile ransomware. In Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Kyoto, Japan, 2–4 November 2015; pp. 382–404.
73. Zhang, H.; Xiao, X.; Mercaldo, F.; Ni, S.; Martinelli, F.; Sangaiah, A.K. Classification of ransomware families with machine learning based on N-Gram of opcodes. *Future Gener. Comput. Syst.* **2019**, *90*, 211–221. [CrossRef]
74. Zhang, B.; Xiao, W.; Xiao, X.; Sangaiah, A.K.; Zhang, W.; Zhang, J. Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes. *Future Gener. Comput. Syst.* **2020**, *110*, 708–720. [CrossRef]
75. Moore, C. Detecting ransomware with honeypot techniques. In Proceedings of the 2016 Cybersecurity and Cyberforensics Conference (CCC), Amman, Jordan, 2–4 August 2016; pp. 77–81.
76. Cabaj, K.; Gawkowski, P.; Grochowski, K.; Osojca, D. Network activity analysis of CryptoWall ransomware. *Prz. Elektrotech.* **2015**, *91*, 201–204. [CrossRef]
77. Damodaran, A.; Di Troia, F.; Visaggio, C.A.; Austin, T.H.; Stamp, M. A comparison of static, dynamic, and hybrid analysis for malware detection. *J. Comput. Virol. Hacking Tech.* **2017**, *13*, 1–12. [CrossRef]
78. Lokuketagoda, B.; Weerakoon, M.P.; Kuruppu, U.M.; Senarathne, A.N.; Abeywardena, K.Y. R-Killer: An email based ransomware protection tool. In Proceedings of the 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, Sri Lanka, 8–11 August 2018; pp. 1–7.
79. Kok, S.; Abdullah, A.; Jhanjhi, N. Early detection of crypto-ransomware using pre-encryption detection algorithm. *J. King Saud Univ.-Comput. Inf. Sci.* 2020, *in press*. [CrossRef]

80. Ashraf, A.; Aziz, A.; Zahoora, U.; Rajarajan, M.; Khan, A. Ransomware Analysis using Feature Engineering and Deep Neural Networks. *arXiv* **2019**, arXiv:1910.00286.
81. Alam, M.; Sinha, S.; Bhattacharya, S.; Dutta, S.; Mukhopadhyay, D.; Chattopadhyay, A. RAPPER: Ransomware prevention via performance counters. *arXiv* **2020**, arXiv:2004.01712.
82. Shukla, M.; Mondal, S.; Lodha, S. Poster: Locally virtualized environment for mitigating ransomware threat. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 24–28 October 2016; pp. 1784–1786.
83. Silva, J.A.H.; Hernández-Alvarez, M. Large scale ransomware detection by cognitive security. In Proceedings of the 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), Salinas, Ecuador, 16–20 October 2017; pp. 1–4.
84. Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. A 0-day aware crypto-ransomware early behavioral detection framework. In Proceedings of the International Conference of Reliable Information and Communication Technology, Johor Bahru, Malaysia, 23–24 April 2017; pp. 758–766.
85. Roy, K.C.; Chen, Q. DeepRan: Attention-based BiLSTM and CRF for Ransomware Early Detection and Classifcation. *Inf. Syst. Front.* **2020**, *23*, 299–315. [CrossRef]
86. Chandrasekar, K.; Cleary, G.; Cox, O.; Lau, H.; Nahorney, B.; Gorman, B.; O'Brien, D.; Wallace, S.; Wood, P.; Wueest, C. ISTR April 2017. *Internet Secur. Threat. Rep.-Symantec* **2017**, *22*, 77.
87. Hwang, J.; Kim, J.; Lee, S.; Kim, K. Two-stage ransomware detection using dynamic analysis and machine learning techniques. *Wirel. Pers. Commun.* **2020**, *112*, 2597–2609. [CrossRef]
88. Al-Hawawreh, M.; Sitnikova, E. Leveraging deep learning models for ransomware detection in the industrial internet of things environment. In Proceedings of the 2019 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 12–14 November 2019; pp. 1–6.
89. Al-Hawawreh, M.; Sitnikova, E. Industrial Internet of Things based ransomware detection using stacked variational neural network. In Proceedings of the 3rd International Conference on Big Data and Internet of Things, Melbourn, Australia, 22–24 August 2019; pp. 126–130.
90. Lee, K.; Lee, S.-Y.; Yim, K. Machine learning based file entropy analysis for ransomware detection in backup systems. *IEEE Access* **2019**, *7*, 110205–110215. [CrossRef]
91. Ahmadian, M.M.; Shahriari, H.R.; Ghaffarian, S.M. Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In Proceedings of the 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), Rasht, Iran, 8–10 September 2015; pp. 79–84.
92. Da-Yu, K.; HSIAO, S.-C.; Raylin, T. Analyzing WannaCry ransomware considering the weapons and exploits. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 17–20 February 2019; pp. 1098–1107.
93. Cusack, G.; Michel, O.; Keller, E. Machine learning-based detection of ransomware using SDN. In Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, Tempe, AZ, USA, 21 March 2018; pp. 1–6.
94. Ahmed, Y.A.; Koçer, B.; Huda, S.; Al-rimy, B.A.S.; Hassan, M.M. A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection. *J. Netw. Comput. Appl.* **2020**, *167*, 102753. [CrossRef]
95. Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Gener. Comput. Syst.* **2019**, *101*, 476–491. [CrossRef]
96. Zuhair, H.; Selamat, A. RANDS: A Machine Learning-Based Anti-Ransomware Tool for Windows Platforms. In *Advancing Technology Industrialization Through Intelligent Software Methodologies, Tools and Techniques*; IOS Press: Amsterdam, The Netherlands, 2019; pp. 573–587.
97. Kok, S.; Azween, A.; Jhanjhi, N. Evaluation metric for crypto-ransomware detection using machine learning. *J. Inf. Secur. Appl.* **2020**, *55*, 102646. [CrossRef]
98. Adamu, U.; Awan, I. Ransomware prediction using supervised learning algorithms. In Proceedings of the 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey, 26–28 August 2019; pp. 57–63.
99. Chen, Q.; Islam, S.R.; Haswell, H.; Bridges, R.A. Automated ransomware behavior analysis: Pattern extraction and early detection. In Proceedings of the International Conference on Science of Cyber Security, Nanjing, China, 9–11 August 2019; pp. 199–214.
100. Pundir, N.; Tehranipoor, M.; Rahman, F. RanStop: A Hardware-assisted Runtime Crypto-Ransomware Detection Technique. *arXiv* **2020**, arXiv:2011.12248.
101. Almashhadani, A.O.; Kaiiali, M.; Sezer, S.; O'Kane, P. A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware. *IEEE Access* **2019**, *7*, 47053–47067. [CrossRef]
102. Bahrani, A.; Bidgly, A.J. Ransomware detection using process mining and classification algorithms. In Proceedings of the 2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), Mashhad, Iran, 28–29 August 2019; pp. 73–77.
103. Poudyal, S.; Dasgupta, D. AI-Powered Ransomware Detection Framework. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, Australia, 1–4 December 2020; pp. 1154–1161.
104. Ahmed, Y.A.; Kocer, B.; Al-rimy, B.A.S. Automated Analysis Approach for the Detection of High Survivable Ransomware. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 2236–2257.

105. Zuhair, H.; Selamat, A.; Krejcar, O. A Multi-Tier Streaming Analytics Model of 0-Day Ransomware Detection Using Machine Learning. *Appl. Sci.* **2020**, *10*, 3210. [CrossRef]
106. Kok, S.; Abdullah, A.; Jhanjhi, N.; Supramaniam, M. Prevention of crypto-ransomware using a pre-encryption detection algorithm. *Computers* **2019**, *8*, 79. [CrossRef]
107. Yang, C.-Y.; Sahita, R. Towards a Resilient Machine Learning Classifier-a Case Study of Ransomware Detection. *arXiv* **2020**, arXiv:2003.06428.
108. AbdulsalamYa'u, G.; Job, G.K.; Waziri, S.M.; Jaafar, B.; SabonGari, N.A.; Yakubu, I.Z. Deep Learning for Detecting Ransomware in Edge Computing Devices Based On Autoencoder Classifier. In Proceedings of the 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), Mysuru, India, 13–14 December 2019; pp. 240–243.
109. Basnet, M.; Poudyal, S.; Ali, M.; Dasgupta, D. Ransomware Detection Using Deep Learning in the SCADA System of Electric Vehicle Charging Station. *arXiv* **2021**, arXiv:2104.07409.
110. Ganfure, G.O.; Wu, C.-F.; Chang, Y.-H.; Shih, W.-K. DeepGuard: Deep Generative User-behavior Analytics for Ransomware Detection. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; pp. 1–6.
111. Nurnoby, M.F.; El-Alfy, E.-S.M. Overview and Case Study for Ransomware Classification Using Deep Neural Network. In Proceedings of the 2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM), Manama, Bahrain, 19–21 November 2019; pp. 1–6.
112. Ullah, F.; Javaid, Q.; Salam, A.; Ahmad, M.; Sarwar, N.; Shah, D.; Abrar, M. Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls. *Sci. Program.* **2020**, *2020*, 8845833. [CrossRef]
113. Qin, B.; Wang, Y.; Ma, C. API Call Based Ransomware Dynamic Detection Approach Using TextCNN. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, 12–14 June 2020; pp. 162–166.
114. Aurangzeb, S.; Rais, R.N.B.; Aleem, M.; Islam, M.A.; Iqbal, M.A. On the classification of Microsoft-Windows ransomware using hardware profile. *PeerJ. Comput. Sci.* **2021**, *7*, e361. [CrossRef] [PubMed]
115. Abdullah, Z.; Muhadi, F.W.; Saudi, M.M.; Hamid, I.R.A.; Foozy, C.F.M. Android ransomware detection based on dynamic obtained features. In Proceedings of the International Conference on Soft Computing and Data Mining, Langkawi, Malaysia, 22–23 January 2020; pp. 121–129.
116. Ahmed, M.E.; Kim, H.; Camtepe, S.; Nepal, S. Peeler: Profiling Kernel-Level Events to Detect Ransomware. *arXiv* **2021**, arXiv:2101.12434.
117. Ayub, M.A.; Continella, A.; Siraj, A. An I/O Request Packet (IRP) Driven Effective Ransomware Detection Scheme using Artificial Neural Network. In Proceedings of the 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, 11–13 August 2020; pp. 319–324.
118. Jethva, B.; Traoré, I.; Ghaleb, A.; Ganame, K.; Ahmed, S. Multilayer ransomware detection using grouped registry key operations, file entropy and file signature monitoring. *J. Comput. Secur.* **2020**, *28*, 337–373. [CrossRef]
119. Alsoghyer, S.; Almomani, I. On the effectiveness of application permissions for Android ransomware detection. In Proceedings of the 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 4–5 March 2020; pp. 94–99.
120. Kim, D.; Kim, S. Design of quantification model for ransom ware prevent. *World J. Eng. Technol.* **2015**, *3*, 203. [CrossRef]
121. Bajpai, P.; Enbody, R. Attacking key management in ransomware. *IT Prof.* **2020**, *22*, 21–27. [CrossRef]
122. Kumari, A.; Bhuiyan, M.Z.A.; Namdeo, J.; Kanaujia, S.; Amin, R.; Vollala, S. Ransomware attack protection: A cryptographic approach. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Atlanta, GA, USA, 14–17 July 2019; pp. 15–25.
123. Lei, I.-S.; Tang, S.-K.; Chao, I.-K.; Tse, R. Self-Recovery Service Securing Edge Server in IoT Network against Ransomware Attack. In Proceedings of the IoTBDS 2020, 5th International Conference on Internet of Things, Big Data and Security, Online Streaming, 7–9 May 2020; pp. 399–404.
124. Monge, M.A.S.; Vidal, J.M.; Villalba, L.J.G. A novel self-organizing network solution towards crypto-ransomware mitigation. In Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg Germany, 27–28 August 2018; pp. 1–10.
125. Mehnaz, S.; Mudgerikar, A.; Bertino, E. Rwguard: A real-time detection system against cryptographic ransomware. In *International Symposium on Research in Attacks, Intrusions, and Defenses*; Springer: Cham, Switzerland, 2018; pp. 114–136.
126. Victoriano, O.B. Exposing android ransomware using machine learning. In Proceedings of the 2019 International Conference on Information System and System Management, Rabat, Morocco, 14–16 October 2019; pp. 32–37.
127. Khan, F.; Ncube, C.; Ramasamy, L.K.; Kadry, S.; Nam, Y. A digital DNA sequencing engine for ransomware detection using machine learning. *IEEE Access* **2020**, *8*, 119710–119719. [CrossRef]
128. Morato, D.; Berrueta, E.; Magaña, E.; Izal, M. Ransomware early detection by the analysis of file sharing traffic. *J. Netw. Comput. Appl.* **2018**, *124*, 14–32. [CrossRef]