Deep Learning Based Hybrid Analysis of Malware Detection and Classification: A Recent Review

Syed Shuja Hussain, Mohd Faizal Ab Razak*
and Ahmad Firdaus

Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia
E-mail: faizalrazak@ump.edu.my
**Corresponding Author*

Received 22 December 2022; Accepted 29 March 2023;
Publication 05 December 2023

Abstract

Globally extensive digital revolutions involved with every process related to human progress can easily create the critical issues in security aspects. This is promoted due to the important factors like financial crises and geographical connectivity in worse condition of the nations. By this fact, the authors are well motivated to present a precise literature on malware detection with deep learning approach. In this literature, the basic overview includes the nature of nature of malware detection i.e., static, dynamic, and hybrid approach. Another major component of this articles is the investigation of the backgrounds from recently published and highly cited state-of-the-arts on malware detection, prevention and prediction with deep learning frameworks. The technologies engaged in providing solutions are utilized from AI based frameworks like machine learning, deep learning, and hybrid frameworks. The main motivations to produce this article is to portrait clear pictures of the option challenging issues and corresponding solution for developing robust malware-free devices. In the lack of a robust malware-free devices, highly

Journal of Cyber Security and Mobility, Vol. 13_1, 91–134.
doi: 10.13052/jcsm2245-1439.1314

growing geographical and financial disputes at wide globes can be extensively provoked by malicious groups. Therefore, exceptionally high demand of the malware detection devices requires a very strong recommendation to ensure the security of a nation. In terms preventing and recovery, Zero-day threats can be handled by recent methodology used in deep learning. In the conclusion, we also explored and investigated the future patterns of malware and how deals with in upcoming years. Such review may extend towards the development of IoT based applications used many fields such as medical devices, home appliances, academic systems.

Keywords: Malware detection, distributed denial of services, artificial intelligence, deep learning, static and dynamic analysis.

1 Introduction

Of malware detection more important while the heavy cost malware attacks are estimated in detecting and rejecting the MALWARE attacks in the domain of network security DNS services. Due to the open use cases available with attracted the research community with a very serious several mobile apps, the mobile-based malware attacks be-attention. Rapid growth in digital technology and their excess of utilization impacts the security of authentication and consequently create hurdles against the safety of personal information. According to recently observed information from cybersecurity report [1], majority of the efforts exerted by the researchers are very steady and monotonous to curb malware attacks. Cyber-attackers never fail to launch campaigns with ransomware, and banking trojans and so forth. majority of cyberattacks includes distributed denial of service (DDoS) which works based on primary attack vector. The research come serious issues to address at large scale. Two types of mobile apps are much popular; one is injected malicious apps another is fraudulent app. Without proper dictionary of the permission declared in the application and API calls, it is a very complex task to address the security against the cyberthreats.

The selection of Android app from huge number of classes, requires high level strategies to maintain higher security

protocols. Alazab et al. [2] discussed an effective malware detection approach which works based on API calls and permission requests. For maintaining maximum likelihood, they divided Android malware into three groups: disruptive, risky, and ambiguous. Malware investigation is used by disaster managers & intelligence officials to: Discover the origin of an assault.

Sort events according to their intensity. Boost the incident management procedure's effectiveness.

The permission-based methods for detecting Android malware are discussed in majority of research works [3, 4] but all the significant contribution discussed in this research, fails to address detecting the variants of obfuscated malware. These classes of malware are much prevalent in evading analysis. Therefore, it become utmost important to develop an approach for removing and detecting such malware from the systems. A detailed documented study on prevention and perception of conventional authentication is discussed in [5]. In several state of the arts, local and global features have been attempted to cope up by introducing code interchange, null value insertion, etc. But very few of them suggested to using hybrid features. Kim et al. [6], proposed a deep generative model by utilizing local and global features from the corresponding binary code sequence and pre-defined latent space of the image converted from malware. The excess of openly accessible resources in Android based smartphones increased the security issues which encourages to spread the ransomware due to its intrinsic infirmity [7]. For both malicious and normal samples, the graphs have been ensured important tools to construct and analyze permission pairs. It is assigned weight to the edge connecting the pairs. From their results, it is observed that graphs-based ransomware detection outperformed several state-of-the-arts on mobile applications used for anti-malware developments. User Account Management (UAC) enables enterprises to instal better-managed desktops & prevents excessive ransomware from harming PCs. Except in cases when an administration has deliberately granted administrator-level privilege to the network, UAC ensures that programs and processes generally operate in the centralized manner of a non-administrator user. A victim must launch an affected System files from the exact location as the intended application for a DLL hijacking to succeed. Criminal groups will be allowed access towards the affected computer anytime it starts if DLL files that are routinely downloaded by programmes are corrupted.

The important branches of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL) have greatly influenced every research domain connected with human profession. The technologies used in Ml and DL have several popular applications such human action recognition [8], security and surveillance, robotics, and many decision-making stochastic process with big data analytic [9]. These technologies are limited to domain specific research [10]. Several applications are noticed with growth of computational advancements in cybersecurity attacks and threats analysis. Some of the

potential applications are noticed and nicely presented in [11, 12]. Deep learning methodology works based automatic pattern recognition from huge amount of unstructured data. From every serious task like medical diagnosis to entertainments [13], deep learning play a crucial role. As every tasks generally follows a pattern but many times it is very hard to find. In such cases, deep learning ensured itself a powerful candidate for pattern identification of ransomware activities [14, 15].

1.1 Malware Categorization

There are several ways to categories the malware based on its nature of detection, pattern, and behaviour etc. Behavior-based or signature-based techniques are the two primary types of malware sensing technologies. Additionally, there are two types of malware analysis – static and dynamic – that are typically used to detect malicious software. Malware detection methods can be generally divided into two groups: anomaly-based identification or signature-based recognition. An anomaly-based detecting method determines whether a software is malevolent by using its understanding of what typical practice entails. The general approach to classify the malware based on category is presented in Figure 1 which shows three basic approaches, type of ransomware, its behaviour, and privilege given to take attention for security. Static and dynamic analyses of infection are the two main categories. Additionally, you might categorise malware analysis according to the amount of work required, choosing between human and automatic study. Although, in the description, we have considered the following global approaches which include misusing the authentication and anomaly detection. The identity of an individual can be threatened based on interpreting the user's signature, misbehaving with system authentication and hybrid mode. Each of the concepts is explained concisely as follows.

1.1.1 Misused authentication approach

Misusing the user's authentication includes signature, behavior, and hybrid approaches as discussed below.

- *Signature based approaches*: The malware detection approach based on signature utilizes the code or patterns stored in the machine. The major drawbacks of this scheme is the system does not remains updated. The periodically updating aspect causes the issues of additional computational overheads [16]. The signature-based malware detection approach discussed in [17] deals with byte codes which suits for mobile devices.

RPackDroid [18] is an android-based malware detection system follows static analysis in supervise mode using API system calls. Zhang et al. also proposed [19] static analysis in fingerprint-based ransomware detection in binary and multi-classifier modes. Another version of ransomware detection system [20], they extracted contextual information in which deep network was designed on N-gram of Natural Language Processing (NLP) opcodes.

- *Behavioral based approaches*: The malware detection based on behavioral data works on the samples collected from machine behaviour and monitors the machine in temporal sequences. The early warning detection scheme is presented in [21] to monitor the user's files. The major limitation of the system was noticed in failure to detect Zero-day attack. Another improved approach follows real-time detection discussed in [22] to carry out the study on locky ransomware. The self-defensible SSD is designed to facilitate extortion identification and restoration without relying on a host-level approach. Another ransomware detection approach discussed in [22] refereed as Self Defensible (SSD) in which the attached storage device was actively monitored for detection purpose. An alarmbased honeypot technique developed [23] manages the folders for monitoring the changes. A sort of deceptive device that enables users to comprehend the behavioural patterns of attackers is called a honeypot. In order to gather information about how hackers behave, security personnel might employ honeypots to analyze cybersecurity risks. Another of the encryption techniques that assists a company in catching infections, spyware, or hackers is the honeypot. It serves as a burglar alarm that identifies efforts to assault a system. The limitation of honeypot folders is lack of synchronization with malware attacks. In [24] Cryptowall ransomware is described where maltester and the folder of Honeypots performed dynamic analysis.
- *Hybrid Detection Approaches*: A hybrid approach jointly presents the features of both misused and behavior-based detection systems. Hybrid approach is described in [16, 25] for android-based mobile device in which misused signature approach given higher priority. An automated deep network-based ransomware detection method is described in [26] for suspicious email filtration and separation. In this approach, Proactive Monitoring System Monitored (PMS) was utilized to detect the ransomware attack. Proactive monitoring in the context of surveillance solutions typically entails spotting possible problems in IT programs

or equipment before found by finding them and complaint, then taking steps to stop them when customers discover them or the problem has an effect on the company. Real-time surveillance of a company's most important IT assets, operations, and data processing is referred to as proactive monitoring. The ability to customise the company's IT remote monitoring needs is based on a thorough knowledge of how the company employs its technology capabilities. Pre-Encryption Detection Algorithm (PEDA) was developed for early detection [27]. PEDA is also able detect Zero-day attack but fails in the attack by self-encrypted code. A hybrid feature engineering were performed in [28] with deep convolution network for developing hybrid ransomware detection system.

1.1.2 Anomaly detection approach

The technique transforms the behavior of anomaly into different normal profiles [29]. The anomaly detection approach utilizes benign to build normal profile. In the local environment, client-server approach used to build file tracker which works as dynamic ransomware [30]. In case of suspicious features notified, the server and user were given the signal of ransomware attack. The complex client-server architecture causes loss of up to twenty files. An important detection and prevention model proposed based on anomaly was discussed in [31]. The model follows a four-phase anomaly detection and prevention scheme which was developed on the basis of unstructured data collected from WannaCry logs and Petya files. These files are available in the directory of EcuCERT institute. A hybrid anomaly based ransomware detection system [32] was developed by integrating the behavioural data and anomaly data. Hybrid IDS incorporates SIDS with AIDS to identify either unexpected and recognized assaults, so overcoming the drawbacks of SIDS and AIDS. The system successfully detected the Zero-day attacks. A deep ransomware detector, DeepRan [33] was developed for monitoring the malicious activity with the help of joint classifier Bidirectional Long Short Term Memory (BiLSTM) and Conditional random Filed (CRF) to distinguish the normal and infected files.

1.2 Research Contributions

- In this review article, our objective is to demonstrate that growing digital technology must have an attention towards the scope and importance

of security issues against safety and security at wide globe. We also add precise representations of up-to-date important stateof-the-arts by ensuring that the potential of Artificial Intelligence (AI) and its recent versions like machine leaning, deep learning etc. can provide a shield against globally warning condition of digital terror.

- We described the different nature of malware (i.e. types, behaviour, and privilege) and its impact on digital services which can attract the researchers to increment the research methodology.
- Keeping the comprehensive reviews on machine learning techniques, we presented a detailed approach with listing the latest studies on static, dynamic and hybrid analysis of malware. The analysis part follows the importance of counteraction measures in malware prediction and prevention.
- This survey paper presented recent picture with outlining detection, prevention, prediction of ransomware research explored from recently published high impact articles on dynamic and static malware detection with deep learning. From the literature, we also we presented the handcrafted, automated, and hybrid feature engineering methods.
- The concise representation of recently used datasets for malware detection along with methodology is also listed. The meta data of database samples includes the sources of availability and tools used for analysis.
- With the extensive overview of the studies carried out on ransomware detection, we have listed pros and cons of the existing studies. Further, with respect to recent technology and the challenging issues raised in history, our review article gives strong recommendation to develop a anti-malware devices.
- Finally, this review article helps the researchers to address the predicting challenges and their solutions.

The remaining structure of the review article follows five more sections from Related Concept to Conclusion. Section 2 represents the descriptive features the nature of ransomware by focusing on static, dynamic and hybrid analysis. Section 3 gives derails of the potentials of AI with respect to feature engineering i.e. handcrafted, automated, and hybrid features. Section 4 pictures the dataset details along with the state-of-the-arts performance on the them. The importance of Section 5 is to address the open challenging issues with existing malware research followed by future recommendations towards building up a robust malware detection system. Finally, Section 6 concludes the the over all summary of this review article.

2 Related Concepts

In this section, we precisely describe the approaches of malware detection based on its analysis. It includes three approaches, static, dynamic and hybrid analysis. The analysis parts follow the subsections describing the prediction and prevention of the malware.

2.1 Static Analysis

Static analysis of malware detection works based the percentage of dangerous function call. It is also applicable in several operating systems. Jacob et al. (2012) [35] proposed static analysis using the features of previously detected malware. The working procedure in static analysis is based vector space model in which string of features are extracted from disassembled Android applications. Then finally, malware is detected by distance metric like Euclidean distance and Manhattan distance. The separation between two points is known as the Euclidean distance in arithmetic. In all other terms, the amount of the connected component connecting two places is what is meant by defining the Euclidean distance among two locations in Cartesian coordinates. The separation of two points determined across right-angled planes is termed as Manhattan distance. If there is significant complexity in the information, the Manhattan distance is generally utilized instead of the more prevalent Euclidean distance. Arp et al. (2014) introduced as an important tool Drebin [36] as the faster detector of malware with static analysis approach. Bayesian classification tools which reverse engineering are discussed in [37] also played important role in android-based malware detection techniques. Static Analysis Module (SAM) is developed by Armando et al. (2014) [38]. Static analysis, often known as unit testing, is a technique for troubleshooting software programs that involves looking at the source before actually running the programme. Pan et al. (2020) [39] presented a detailed analyses of static approach of malware detection. Recently, satisfactory solutions were suggested for sounding problem 'sheer protection' which does not allow them to attains its priory information. Syrris et al. (2021) [40] resolved the sheer protection problem by utilizing the popular machine learning classifiers. Determining whether an app is malware or is a very tedious task. Idrees et al. (2017) [41] correlated the intents and access permissions with ensemble learning to deduce the problem with 99% accuracy. The same solution was provided in [42] with testing the effectiveness of Bayesian Network(BN) on dataset consisting of 5,560 malware apps and 1,846 goodware and indicate 95% true positive rate. The Opecode sequence of malware file can be utilized

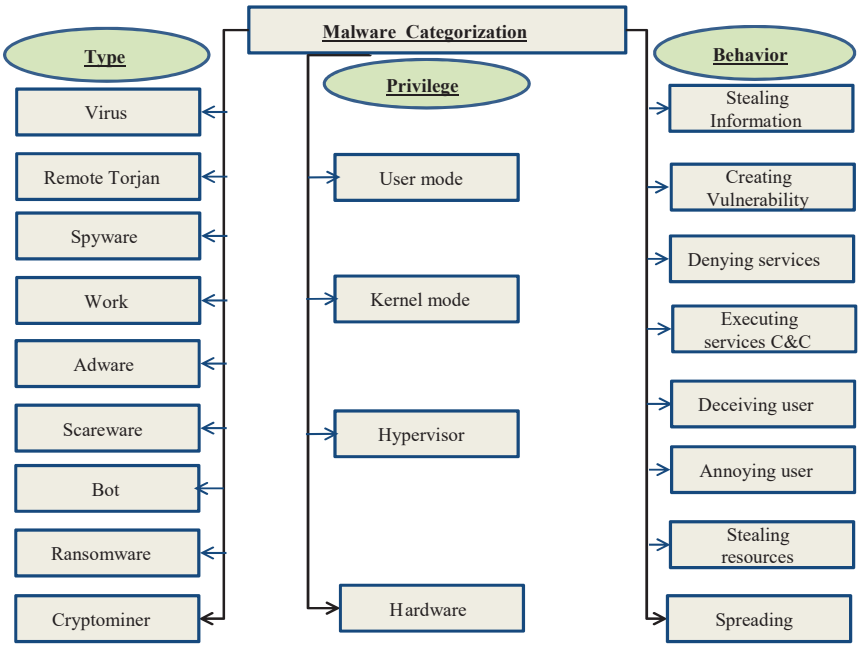

Figure 1 Categorization of malware classes [34] and its nature.

as they do not need any support of virtual box system. Therefor, Zhang et al. [19] proposed a unique static method by using opcode sequence into the sequences of N-gram files. They noticed the drawback of their opecode extraction put an obvious mark on validity of real dataset.

2.2 Dynamic Analysis

The main issue with the technique of static analysis-based malware detection is the weakness against the evading of signatures where zero-day attacks and code obfuscation are major concern. The major characteristics of dynamic analysis of malware detection techniques observe the behavior of malware during execution mode. Dynamic analysis examines the behaviour & activities taken by the application when it is running to determine whether it is infection or not. Both approaches have merits and drawbacks. Infections using ransomware can result in expensive business interruptions as well as the destruction of key data. From the informative concept of malware factory, You et al. [43] and Lipovsky [44] ensured that unpacking of unique variant results the malware in self-morphing with high payload on its every

execution. As the static analysis-based techniques work based on API calls, it becomes highly obfuscated. These approaches can be easily evaded by utilizing the pattern of API call in ransomware. The methods based on the sequence and patterns of API calls play with important features for dynamic analysis. Although dynamic analysis collects variety of options available on the operation of software, statically study relies on characteristics that are collected beforehand (or emulation). Static analysis is typically more instructive and more effective than vibration loading, especially when dealing with heavily obscured software. Gupta et al. [45] and Youngjoon et al. [46] discussed the sequence of API calls as a string data and for detecting the malware they applied applied string matching algorithms. They selected 534 API calls and categorized them into 26 classes by assigning them an English alphabet. Young et al. [46] also used English alphabets to assign the API function and represented the sequence of API calls of ransomware pattern as a string data, and then applied the DNA sequential alignment algorithms to detect and classy the malware.

The concepts of the similarities of API calls is also important to consider as jaccard measures and cosine similarity gave better performance. Accounting the behaviour API calls, Peisert et al. [47] observed the sequence of malware files and investigated the code sequence responsible for cause of abnormal behaviour. Qiao et al. [48] created cluster of ransomware samples by computing the parameters related to API-calls and corresponding return values of API names. Run time embedding of hooking codes helps to obtain.

API sequences. Then hooking code catches the target executable files which support to protect itself by preventing the hooking API. Alternatively, for preventive measures, ransomware blocking is also successful process. This approach is followed by Bayer et al. [49] and developed an scheme called TTAnalyze which do not require the modification in executable files and efficiently extracts the API sequences to block the avoidance attempts of ransomware. Another approach for malware detection can be considered based the encryption behaviour of malware. The techniques developed for CryptoLocker system can efficiently detect the encryption behaviour of the malware. An example of malware is CryptoLocker, which locks down affected machines through scrambling their data. When attacked, users are required to submit a "extortion money" to have their data decrypted and restored. In the functioning of CryptoLocker system, it is enabled alert until the ten files are lost. Monitoring the real time updates in the malware files and the suspicious behaviour is recorded with alert by CryptoLocker system. Ransomware attacks combine symmetrical and asymmetrical encrypting

methods for better efficiency. The hybrid encryption strategy enables intruders to swiftly secure their perpetrator's information without jeopardising the safety of their system. System behaviour on high alert suspends all the suspicious updates to protect the system. Windows operating based API calls for malware detection is a quite popular technique. Sgandurra et al. [50] extracted 30,000 feature sets based on several parameters file operations and extension, key of registry and windows API calls. All these extracted features are processed with regression classifier. The technique of turning unprocessed numbers into numerical characteristics that can be handled whilst keeping the data from the source given dataset is referred to as feature extractor. Kharraz et al. [51] and Butler et al. [21] jointly detected ransomware by analyzing the encrypted behaviour of ransomware at massive scale. The only difference between both these techniques is the observation methods, Kharraz et al. is based on I/O file requests whereas Butler et al. used the updates in entropy of malware files. In [51], maintaining the Master File Table (MFT) is main system overhead in detecting the encrypted actions in the files. The better maintenance of MFT results in higher performance of ransomware detection. To monitor the updates in file systems, Continella et al. [52] helps in finding specific ransomware activities by developing a protection system called Shield File System (ShieldFS). In the closed sense of malware detection, Kolodenker et al. [44] proposed the recovery system after ransomware attack is confirmed. Their system works based on the storing the encryption keys which is later used to recover the infected files.

2.3 Hybrid Approach

The increasing harmful impacts on every upcoming technology, motivates the researchers to face open challenges in malware detection, classification and its prevention to ensure the security parameters. Nearly each element of modern life is impacted by technologies, including productivity, socializing, food and healthcare accessibility, and transportation effectiveness & security. The influence of the internet has facilitated the emergence of international groups and made it easier to exchange resources and data. The difficulties that malware identification research has encountered have been identified as imbalanced data, accessible & publicly standards, idea drifting, antagonistic training, or modeling understandability. The categories of malware detection based on feature modelling are considered as global, local and hybrid. Several features like hex, dump or extracted from disassembled files are discussed in [53] for showing distinctive characteristic of various malware groups.

State-of-the-art approaches presented in [54] for clustering Android malware rely heavily on the raw labels from commercial AntiVirus (AV) vendors. Chen, X. et al., proposed a new approach to Android malware clustering that embedded all malware in the network into a low-dimensional and compact hybrid feature space for effectively clustering weakly-labeled malware. ANDRE is a new and Hybrid Representation Learning approach to clustering weakly-labeled Android malware. It uses a three-layer Deep Neural Network to partition the known and unknown malware. A characteristic related to data information or behaviour is needed to demonstrate the items, including document analytics or a listing of the API methods that were employed. Additionally, every item is linked to the designated appropriate response. ANDRE achieves comparable accuracy to the state-of-the-art approaches for clustering ground-truth samples. For instance, people are aware from experiences that you're more likely to overlook the symbol users are aiming to select when users move their pointing device quickly. According to this, you are aware that you must proceed slowly whenever opening the entrance to make certain the key engages the locking.

Chen, L. et al., deliberate the spread and detection of mobile malware on a large scale by studying SMS/MMS, Bluetooth, 5Gbase station networks, metropolitan area networks, social networks, telecommunication networks, and application market ecosystems. Through research, author obtained global malware threat detection, traceability, and propagation models.

2.4 Counteraction Measures for Malware Detection

Richardson et al. [55] discussed an overview of detection and classification of malware with concise measure of prevention techniques. Looking towards the irreversible nature of malware attacks requires the early prediction of security issues. Although, the dynamic approach can protect against the evade due to code obfuscation techniques, The early detection requires to understand the behaviour and pattern of the malware. Several recent techniques are not found much satisfactory to based on the information stored in database. Students' performance in the discipline is likely to suffer if instructors do not have the training, resources, and fundamental topic knowledge and abilities to deliver or execute the media studies program. Therefore, malware prevention is pointed highly preferable [56, 57]. Several studies put efforts to bring the clear state-of-the-arts of malware analysis. All of them include three basic parameters, prevention from malware,prediction of security,and detection of malware [29, 58] which is explained in following subsections.

2.4.1 Prediction

The objective of predicting the malware aims to stop the attack before it is taken placed at machine. The very first step step is taken to stop occurrence the undesired process. It is bring into picture by gathering information suitable to predict the possible attack [58]. An android based ransomware prediction and detection system was proposed In [59] androidbased ransomware prediction system is proposed which ensure the scanning of all the apps using permission call. Data mining and Machine learning techniques are utilized for predicting ransomware attack [60]. With honeypot approach, out of 30,000 attributes, only five attributes are considered to implement six algorithms of machine learning. In this publication, we outline the six phases that go into creating a machine learning modeling: information retrieval & gathering, sample selection and investigation, modeling building and training, evaluation metrics, launching innovative, then prototype maintenance.

2.4.2 Prevention

The objective prevention aims to avoid the situations in ransomware attacks are occurred. Line charts, visualizations, scatter plots, descriptive statistics, or scatter diagrams are the most typical varieties. It also targets fixing securities reported in the previous versions. In the following consequence, prevention studies are further categorised into two classes, Reactive studies and Proactive studies [61].

- *Proactive Prevention*: In [62], a preventive measure was developed to monitor the updates in directories and process used in mobile based ransomware detection. The abnormal or infected process shows different statistics and force to terminate the system. The faster technique enable to detect new ransomware attacks.

The model presented in [63] extracts the payloads from real-world traffic and deep learning methodology is used for early detection and prevention of ransomware. An early detection prevention system proposed in [31] accomplishes the task of prevention with unstructured data collected from Wannacry and petya. Further by collecting the relevant features of ransomware patterns, deep learning was adopted for predicting the attacks. The preventive measures presented in [64] aborted the encryption process before the attack. They utilized the exchange approach for monitoring and breaking the connection respectively the process is normal and malicious. An important proactive prevention-based ransomware model was presented in [65] by performing hybrid analysis.

- *Reactive Prevention*: The prevention approach aims at the next target of Proactive prevention. The reactive prevention is very closed to ransomware detection in which major focus is put on recovery of loss when the ransomware has been taken placed. In [66] reactive prevention technique is presented for detailing the data recovery and back up after the attacks. In their methodology, frequent write and read request are performed on storage or Self Defensible Devices (SSD). RAPPER, a tool for detecting ransomware is proposed in [67] which took two steps, one is analysis of statistics of the process and another is detection of ransomware. This framework composed of LSTM and Fast Fourier Transform(FFT) to manage back up if attack occurs. The main objective of reactive prevention is to have supervision over data control flow. Data Acquisition and Supervisory Control (DAASC) system [68] ensured higher impact.

3 AI in Malware Detection

3.1 Handcrafted Features

The importance of achieving key feature components of malware remains at higher priority but varied range of parameters in the feature set is trivial problem to classify the feature map of malware. Infected computers, blackmail, worm, trojan horses, as well as malware are examples of common malware. These harmful applications have the power to take away, encode, or erase private information, change or take over crucial data processing, and keep an eye on the perpetrator's web behavior. A simple issue is simple to resolve: It is not easy to teach computers to comprehend natural speech. It was a badly sourced program that raised unimportant issues while ignoring more prominent accomplishments. According to Zhang et al. [54]. The process of extracting features is categorised into two steps. One is based on code similarity analysis (CSA) and another uses manifest files of Android analysis. In CSA approaches, pairwise comparison scheme called ANDRE is adopted which dissects the similarity between the android apps. Removable assets used frequently in Android applications are third-party frameworks. Although third-party modules offer a wide range of capabilities, they also pose significant private or safety concerns. By considering two input apps at a time, ANDRE scheme yields similarity file for all pairs. The advantages of similarity file is taken in computing the similarity score for the source code files. In addition, the size of index can also be reduced by filtering heuristics.

The major portion of an Android app is covered by libraries of safe third party which is termed as noise in representation learning [69]. In CSA, the noise of code segments related to libraries is removed from the application code of malicious app by whitelist standards. The major advantages of whitelist are in adding new APIs in very flexible and safe modes. Another approach of extracting feature is based on analysis of manifest files. The meta-information of the malware apps include package name, permissions, and API versions etc. The manifest files are useful to extract the meta-information as they are connected with corresponding network. In this way, the ground-truth files are stored with their uniquely associated family names.

Wang et al. [70] discussed a framework to detect malware apps and categorize the benign apps with the ensembles of five classifiers. Author describe an alarm system that would be triggered when app is identified the malicious. In the experiments of malware app detection, and ensemble method to achieve the detection accuracy as 99.39%. The experimental results show that proposed ensemble method is more robust than the five base classifiers in the detection and categorization. Malware presents a serious threat to the security of devices and the services they provide. Zhu et al. [70] proposed SEDMDroid: a static malware detection framework, have two-tier architecture, including the ensemble of MLP and the fusion of base leaner output by SVM. They conducted experiments on two different datasets to verify the effectiveness of the proposed method. Xu et al. [71], proposed an Android-COCSO, a supervised approach for detecting Android malware. It was used to combine both byte-code of the DEX file and native code of the original file. The performance of large-scale experiments on 100,113 samples (35,113 malwares and 65,000 benign) show that the Android-COCSO approach detected malware applications with an significantly outperforming accuracy of 99.86%, to meet better state-of-the-art solutions.

3.2 Automated Features

Deep Learning approaches already have been popular on image classification [72] and the task of several other important task like machine translation [73] and text classification [74]. The advantages DL are obvious learning the complex features automatically via stacking of hidden layers [75]. In the deep networks, there are multiple level of abstractions which promote the automatic learning in efficient fashion. In this way, the DL techniques can easily identify the latent features which are very hard for the expert to represent [76]. In android app, DL based methods are highly suitable to capture

semantic information. This is particularly assumed that semantic data contains huge amount of information to train the deep network. Recently, Feng et al. [77] introduced research state that Android malware detection is mostly performed on server side. These consist of endpoint infection identifying and responding, computer learning-based dynamic identification, or software keyword filtering. Methods for deep learning identification can be employed to recognise and distinguish between good and bad data. MobiTive is a pre-installed solution rather than an app scanning and monitoring engine. MobiTive is a performance-sensitive Android malware detection system on mobile devices as a preinstalled solution. It uses customized deep neural networks to provide a real-time and responsive detection environment on mobile devices. It can provide a reliable detection accuracy and fast responsive (i.e., less than 3 seconds on average) detection service on mobile devices directly. Android has become the most popular mobile intelligent operating system. However, more and more attackers take Android as the primary target. Wang et al. [78], designed an Android application classification model based on multiple semantic features. Key features help identify dangerous behaviours in unknown applications more effectively. Author also investigate to combine input generator tools Intelli-Droid and Droid Box to improve dynamic analysis coverage. The results demonstrate that the detection accuracy of malware is 99.39%.

3.3 Hybrid Features

The hybrid approach is concerned with solutions of both dynamic and static analysis of malware detection and classification. The issues raised with conventional handcrafted feature mechanism for malware detection directs towards AI based feature learning. Several methods [79, 80] have been proposed which automatically analyze the features of malware. For example, Peng et al. [81] used probabilistic learning approach. Another similar approaches for developing malware detector followed machine learning classifications and extracted features with android applications DroidAPIMiner [82], Crowdroid [83], and MAST [84]. AI based algorithms have been popular with with experiments of machine leaning and deep learning.

With popularity of deep learning methods, Kalash et al. [85] and Vasan et al. [1] proposed a CNN-based malware classification architecture. The malware binaries files used in experiment were first updated into gray-scale images to give input to CNN layer. Image based machine learning approach

for malware detection are found much suitable and outperformed the popular state-of-the-arts. Gibert et al. [86] followed agnostic deep training network on Microsoft Malware Classification Challenge (MMCC) and MaI-IMG datasets. Marastoni et al. [87] adopted syntactic code transformations with CNN methods for developing semantically persevered malware detection system. The outcomes of their proposal was bi-directional long short term memory (BLSTM) and CNN. Both the networks were trained on the features extracted from the images of the generated dataset. Transfer learning, a part deep network technology is recently introduced in many vision-based applications. Bhodia et al. [88] encouraged the researchers for malware detection and classification by transfer learning on image-based approach. Before applying transfer learning on recognition model, the executable files first converted into images. Prajapati et al. [89] also considered the importance the features extracted from malware images. In their experiments, multiple neural networks like Vanilla recurrent neural network(VRNN) introduced in [?] and multi-layer perceptions were used on very diverse datasets. They also performed the complex experiments by combining Gated recurrent Unit(GRU) and LSTM. However, similar experiments were also performed by Pei et al. [90] for transfer-based semi-supervised machine learning to detect IoT enabled malware. Yajamanam et al. [91] implemented image-search based global image descriptor (GIST) descriptor proposed in [92]. After analysing and testing the GIST descriptor for malware detection, they compared it with deep neural networks algorithms. Vasan et al. [93] explored and designed CNN-based architecture with ensemble learning which demonstrated the accuracy of 98% on packed malware detection. CNN with extreme learning machine(ELM) approach were adopted by Jain et al. [94] to design malware classification system. the malware detection system of ELM and CNN performed satisfactorily on 1-D and 2-D applications.

4 Datasets and Detailed Analysis

The highly noticed issues with unavailability of dataset is unpredictable behaviour of malware. Some of the very popular dataset published and accepted to perform standard experiments are listed in Table 1. In addition, many datasets are not publicly available and bear the format issues. In such situation applying machine learning or data mining techniques is quite difficult. For the convenient experiments on malware detection, Table 1, presents state-of-the-arts and meta data along with detailed explanation.

Table 1 Dataset family used for ransomware detection and classification

State-of-the-arts	Platform	Dataset-Class	Dataset-Family	No.Samples	Source
MRMR [95]	Windows	Ransomware	–	1354	VirusShare and VirusTotal
		Benign	–	1358	Software-informer System
RANDS [96]	Window	Ransomware	Reveton	400	VirusTotal Malware Blacklist
	Window	Ransomware	WinLock	2620	VirusTotal Malware Blacklist
	Window	Ransomware	Archiveus	1500	VirusTotal Malware Blacklist
	Window	Ransomware	CryptoLocker	720	VirusTotal Malware Blacklist
	Window	Ransomware	RaaS	310	VirusTotal Malware Blacklist
	Window	Ransomware	CryptoWall	3250	VirusTotal Malware Blacklist
	Window	Ransomware	AiDS	400	VirusTotal Malware Blacklist
	Window	Ransomware	GpCode	800	VirusTotal Malware Blacklist
DRTHIS [97]	Fog Layer	Ransomware	–	660	VirusTotal
		Benign	–	219	–
RanSD [28]	–	Static	Ransomware	219	VirusTotal VirusShare
	–	3646	Goodware	1700	Window7
	–	Dynamic	Ransomware	1946	VirusTotal VirusShare
	–	3444	Goodware	1455	Window7
BilSTMDeepRan [33]	Networks Bare metal server	Ransomware event logs	–	17	PC host logs
		Benign event logs	–	103,330	PC host logs

(*Continued*)

Table 1 Continued

State-of-the-arts	Platform	Dataset-Class	Dataset-Family	No.Samples	Source
ESRS [98]	–	Ransomware	–	8152	VirusShare
	–	Benign	–	1000	Informer.com
PEDA [99]	Windows	Ransomware	–	491	VirusTotal
		Benign	–	942	Sgandurra
MCPS [100]	Ransomware	–	Wannacry	50,537	–
		–	BadRabbit	50,537	–
MCNC [101]	Ransomware	–	Wannacry	–	–
		–	BadRabbit	–	–
MSWR [102]	Windows	Ransomware	Wannacry	80	–
		Non-Ransomware	BadRabbit	80	–
MSWR [103]	Windows	Ransomware 666	TeslaCrypt	348	VirusTotal
			Unlock26	3	
			WannaCry	1	
		Benign	–	80	Software repository web

Table 2 represents the important observation of studies carried out from the perspective of performance measures of ransomware classification and detection. It can be observed from the literature of ransomware detection, very few experiments are found to map all the parameters of performance measures. Here, we selected the important research components which give straight though insight to the readers for improving the research states. As per our discussion, it is up to date information of the recently performed experiments on ransomware detection since 2016 to 2022. We selected the popular classification measures namely precision, recall, false precision rate (FPR), F1 score and accuracy.

The advancement of AI based technology remains popular in every filed. For taking care of studies of various research domains, Table 3 presents pros and cons of deep learning models used in recently published articles on malware detection. Also Figure 2 for supporting the researchers gives the summary of features extraction methods used for malware detection. With exploration same line of research from the literature on ransomware detection, we presented the popular and highly cited research articles in Table 4. It is listed the key components of the research methods with their corresponding author to extend the technologies and incremental research quickly.

Table 2 Machine learning based recent state of the arts on malware detection and classification

S.N.	Research Study	Precision	Recall	FPR	F1 Score	DetectionRate
1.	RansomWall, Shaukat et al. (2018) [104]	99.94%	97.28%	0.056%	98.84%	98.25%
2.	RansReview, Aurangzeb et al. (2017) [105]	71.19%	88.76%	38.00%	93.92%	88.99%
3.	SDN, Popli et al. (2019) [106]	87.44%	85.14%	12.5%	87.20%	87.00%
4.	DPBD-FE, Rimy et al. (2020) [107]	97.90%	97.10%	2.09%	97.10%	99.83%
5.	API-Malware, Kumar et al. (2017) [108]	97.90%	98.01%	1.00%	99.00%	98.00%
6.	DNAact-Ran, Khan et al. (2020) [109]	89.70%	87.90%	10.00%	88.80%	87.90%
7.	RansHunt, Hasan et al. (2017) [110]	97.88%	97.04%	2.1%	97.49%	97.10%
8.	NetConverse, Sharmeen et al. (2020) [111]	98.38%	97.04%	1.6%	97.74%	97.10%
9.	LSTM, Kok et al. (2019) [112]	NA	NA	3.33%	NA	96.67%
10.	RDML, Bae et al. (2020) [12]	99.40%	99.35%	NA	99.97%	99.53%
11.	GIN (Xuan, et al. (2022) [113]	70.50%	86.32%	NA	77.61%	89.77%
12.	TSRD, Hwang et al. (2020) [114]	NA	96.65%	6.93%	97.40%	98.80%
13.	UNVEIL, Kharraz et al. (2018) [115]	98.34%	97.13%	1.64%	97.72%	97.18%
14.	EldeRan (Sgandurra, et al. (2016) [50]	99.83%	96.33%	0.16%	98.05%	96.36%
15.	Talos (Cimitile, et al. (2020) [56]	97.50%	95.40%	0.28%	97.00%	99.06%
16.	ConRec (Malik, et al. (2022) [116]	82.80%	95.40%	NA	88.85%	99.56%
17.	FSML Zhu, et al. (2022) [117]	85.30%	88.70%	NA	86.20%	98.20%
18.	MGTEnsemble Ahmed et al. (2022) [118]	99.90%	98.00%	NA	99.00%	NA
19.	PEHeader Manavi et al. (2022) [119]	84.72%	84.70%	NA	96.76%	96.80%
20.	DeepAMD, Imtiaz et al. (2021) [120]	93.50%	93.40%	NA	93.20%	93.40%

Table 3 Selected deep learning methodologies for ransomware detection and classification

S.N.	Deep Learning Technology	Pros	Cons
1.	Deep Neural Networks (DNN): Extended version of shalow neural networks, having more than three layers. Saxe et al. (2015) [121] developed deep neural network for security analysis.	Robust to solve many unconstrained problems.	In case of unsupervised learning, without data samples, these are hard to design
2.	Restricted Boltzmann Machines (RBM): The advantages of RBM is used as generative models for statistical analysis in which various races of data are taken as input for likelihood dispersion. In general, RBMs are used to deal with high dimensional temporal data like video, sounds. In [122–124] for static RBM and deep RBM architecture developed malware detection.	Easier data distribution process, better feature extractor for training the model. - It can be used as features extractor to train other models on top of it	Computing consuming and complex training process.
3.	Convolutional Neural Network (CNN): It generally comprises computational three layers with the objective to reduce the number of parameters without focusing on compression of algorithms. Three in layers in CNN are convolutional, pooling, and classification layer. Polling layer and classification layer are also referred as sampling and fully-connected layer respectively. All theses layers utilizes the weight sharing mechanism. The convolution operation basically is applied on the grid of image with the help of synthetic image called filters. Pektacs et al. (2017) [125] used CNN for ransomware classification.	It can explore optimal features with the fewer neuron connections in feed-forwarding processing. Huge class of CNN provide the extensive set of variation for advance research.	Without multiple layers, CNN can not explore better feature space of the visual information. Huge amount of ground truth required to classify the particular image exactly.

(*Continued*)

Table 3 Continued

S.N.	Deep Learning Technology	Pros	Cons
4.	Deep Belief Network (DBN): It is a very sophisticated network made up of various stochastic layers and hidden variables and improves the issues noticed in classic neural network. The main issues are parameter tuning at local minima, slow learning rate and the need of large number of training samples. DBN is basically suits for unsupervised stochastic learning. Yuxin et al. (2019) [126] proposed malware detection based on deep learning in which DBN is used.	Being a generative deep network model, it can generate samples from the auto extracted features which can be used to train the model.	DBN regrets in the use of the applications where 2D spatial information is used such as image processing applications with computer vision problems.
5.	Recurrent Neural Network (RNN): It is a very sophisticated network made up of various stochastic layers and hidden variables and improves the issues noticed in classic neural network. The main issues are parameter tuning at local minima, slow learning rate and the need of large number of training samples. DBN is basically suits for unsupervised stochastic learning. Garg et al. (2022) [127] proposed malware detection with its vulnerability using text processing for Android.	In the computation with RNN model size dies not increase with input since computation are utilized from history.	Slower computational issues is a serious problem with RNN. Inform from the long history is lost.
6.	Long Short Term Memory (LSTM): The LSTM networks [128] are the special case of RNN which can learn to avoid long term dependency scheme. In these sequential networks, the information remains sustained by the mechanism of storage unit reffed as Gates. There are three types of gates in LSTM: Input, forget and output gates. The applications of LSTM can be seen in several android based malware detection techniques [129].	The different gates makes LSTM architecture outperformed in forecasting non linear time series data.	LSTMs are very advanced models to deal with small dataset. Hence, overfitting problem remains unsolved even after adding better regularisation.

(*Continued*)

Table 3 Continued

S.N.	Deep Learning Technology	Pros	Cons
7.	Deep Autoencoder: The architecture of deep autoencoder contains two, symmetrical DBNs with four to five shallow layers. The network is separated for encoding and decoding. The building blocks of layers of DBNs composed of Restricted Boltzmann Machines(RBM)as discussed in [122]. The application of auto-encoder is effectively seen to deal with dimensional reduction, anomaly detection, and image search. Kim et al. (2018) [130] proposed zero-day malware detection method by utilizing Transferred Generative Adversarial Networks(TGAN).	A deep encoder can perform with non-linear transformations of video, image and any time series data efficiently. Its better than PCA in learning with several auto-encoder layers instead of using a highly dense transformation layer. Auto-encoders are nicely implemented with transformer model [131].	In case of insufficiency of data and high compression, encoders are the worst candidates to deal data science and cybersecurity applications.

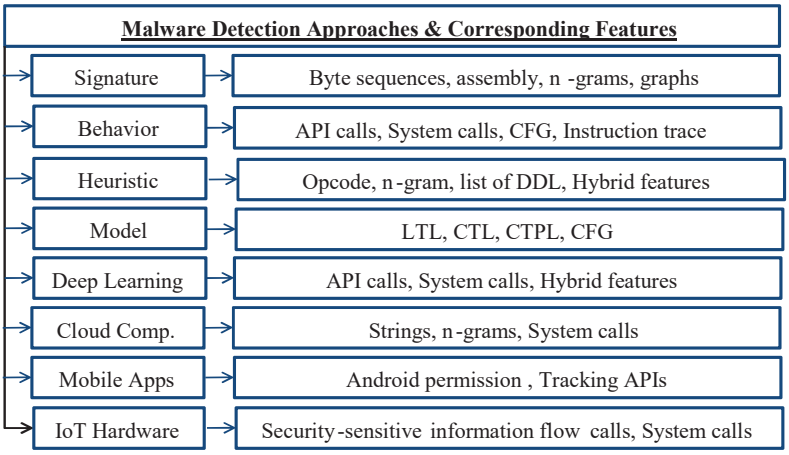

Figure 2 Existing malware detection approaches and corresponding features.

5 Open Challenges and Future Research

5.1 Open Challenges and Limitations

In the previous sections, several important studies on ransomware detection, classification and prevention are presented. Due to unpredictable pattern of

Table 4 Selected studies of feature analysis and future recommendation

S.N.	State-of-the-arts	Performance	Shortcomings	Recommendation
1.	R-Locker, Gomez et al. (2020) [132]	Instant blocking and removal	Android and Window not tested	Honey file not secured
2.	NgramOpocode, Zhang et al. (2019) [19]	Best recall 99.83%	Fails in dealing with locky and crypto wall	Slower API comparison
3.	MFRB, Pektacs et al. (2017) [125]	High accurate Confidence Weight (C = 4.0, n = 6, TrainAC = 94%, TestAC = 91.8%)	Not mentioned	Detection issues with virtual clock
4.	DDefender, Alshahrani et al. (2018) [133]	A user friendly malware DDefender identifier is developed.	All the events are not generated by Monkey tool in dynamic analysis. Overhead in APK upload is noticed.	System must be load balancing.
5.	DRTHIS,Homayoun et al. (2019) [97]	LSTM-Binary (F-measure: 99.60%, TPR: 99.20%, FPR: 0%, MCC: 98.60%) and Multiclass(TPR: 97.20%, FPR: 2.70%)	Advanced Deep networks not discussed	Monitoring Ransomware
6.	NIDS, Hamed et al. (2018) [134]	Accuracy weighted observation bi-gram (accuracy:82%, weighted Recursive Feature Addition (RFA): 92.90%, Joint: 87.80%),	Advanced Deep networks not discussed	N-gram technology with Ensemble classifiers.
7.	LMDM, Kumar et al. (2019) [135]	Decision tree, LDA accuracy on raw features (Accuracy 98.4%)	Real-world situations with malware and benign ignored	Multimedia applications and device. Feature space variation needs to extend. %
8.	MIL, Stiborek et al. (2018) [136]	Accuracy with SVM (94.4%)	Run-time is more (5 mins)%	
9.	VTCMalware, Lin et al. (2018) [137]	Virtual time control Sandbox-8x runs faster @102s (TPR: 98.6%)	Not mentioned	Faster temporal detection
10.	KuafuDet, Chen et al. (2018) [138]	Sophisticated attacker accuracy (96.35%)	Run-time (Slowe: 3 min), More efforts in balancing the accuracy metric.	Reinforcement learning to prevent APK following reverse process.

11.	Talos, Cimitile et al. (2018) [56]	Highly satisfactory (Accuracy: 99.0%)	Triggers are not verified in beginning	Malware like rootkit and botnet with structure similarity measures.
12.	RansWinAPI, Hampton et al. (2018) [139]	API developed with better variance	Not mentioned	Detection mechanism violated.
13.	PADMT, STIborek et al. (2018) [140]	Malware (HomoScore: 76.70%, Complete: 60.90%, V-score: 67,90%) Joint legitimate and malware (HomoScore: 76.10%, Complete: 52.30%, V-score: 62.0%)	Resources monitored samples without sandbox are dealt	Faster run-time limit by 20 sec.
14.	Deep4MalDroid, Hou et al. (2018) [141]	RNN and LSTM networks used on the features generated by Permissions-events of Monkey Tool.	Running apps in the emulator does not revoke to access of showing the malicious activities.	Running time of Apps must be increased.
15.	PEDA, Kok et al. (2020) [99]	It is viable detecting the cryptoransomware before encryption without sandbox	Challenges with database used	Signature repository of multiple languages.
16.	Fesa, Fernando et al. (2022) [142]	Commonly used genetic and swarm intelligence based feature selection methods predict the stochastic system. Developed feature selection architecture to enhance the longevity of classifiers.	Restricted TPR as a detection rate although it may get disturbed in absence of proper data wrangling.	Feature extraction directory can be enhanced by ensemble.
17.	Fesa, Ahmed et al. (2022) [118]	The static network-based features ensemble models are designed for dynamic and static analysis, robust to adversarial evasion.	Ensembling is not easier to interpret as a results it affects the prediction of model performance.	Cost-effective and malicious features for robust validation for the performance of countermeasure of protection.
18.	PEHeader, Manavi et al. (2022) [119]	No software and pre-processing assistance, only headers of executable ransomware file can detection efficiently.	Experiments were performed on very limited dataset.	Recommend to extend the parametric evaluations at higher complexity.

ransomware, it is not acceptable that any state-of-the-art is fully satisfactory to produce a malware-free secured and robust system. In this subsection, we summarize the following open challenging issues which can be targeted to develop a more secured malware detection system.

(1) Cyberattackers put businesses in a situation wherein ransomware is the quickest and least expensive option to recover accessibility over their data through encryption such documents and requesting a ransom amount for the decryptor. Unlike virus, which gives hackers complete exposure to their networks, ransomware simply prevents users from accessing private and sensitive data till a ransom payment is made. With the rapid growth of different nature of ransomware, it is very monotonous task to develop ransomware studies with the help of various logs and the families of ransomware Table 1.
(2) It is very difficult to detect the ransomware which is itself developed by encrypting its own file.
(3) The pseudo or real events, all generate synthetic database by extracting the specific features of some fixed sources. Therefore, absence of dataset is highly discourage the researchers.
(4) The majority of redundant and irrelevant system calls degrade the detection performance.
(5) Pre-encryption early detection may be very helpful source to detect the ransomware. It is found very few literature that follows the concept of population drifts.
(6) The time used in analyzing phase for detecting the sample seriously affects the performance required to maintain the implementation of the detection studies.
(7) Many existing studies available on the analysis part of ransomware is not direct to perform he experiments. Majority of the studies are empirical which do not provide essential report to realize the product outcomes.
(8) From the existing studies, it is most common challenge to notice that determination of the pattern of ransomware starts after it has been encrypted.
(9) Lack of studies available for building the back up system for ransomware detection is serious hurdle on record.
(10) Accessing of Honeypot folder, generated by Honeypot method does not assure the possibility of attack. Since it is not fully secured.
(11) Almost the existing ransomware detection approaches developed a fully platform dependent system which can not be implemented for Windows

and cloud based devices in parallel. Therefore, developing a separate system force to bear the extra cost.

(12) The sustainability of ransomware detection system is quite poor in case of dealing with the data having highly redundant information.

(13) Malware detection system developed with deep learning does not provide transparent interaction how the decision is carried out. When not adequately taught, machine learning runs the danger of using ineffective techniques and producing few forecasts. In order to recognize abnormalities or recognize security concerns, machine learning systems must be educated to evaluate relevant data and form inferences. Also, it is noticeable that deeply learned malware detection cannot be ensured to maintain the performance with new sets of training samples.

5.2 Future Research

In our research paper, we presented a concise representation of the review work on ransomware detection in which machine and deep learning technologies are exploited. After detailed studies of the existing literature, some important research directions are highlighted to space the room for development in the research of ransomware detection and classification. For clear remarks on open issues that need further research on ransomware detection systems is the key concern of this paper. Following are summarized points to measures the importance of research on ransomware detection and classification.

(1) Computational issues and hardware complexities: Maintaining the computational overhead, malware detection system ensures its stability. Detecting ransomware attack in almost real-time is preferred. Therefore, the computational complexity of the system must be very low especially in case of hybrid IoT based secured system. Apart from computational issues with software, hardware complexity also play an important role in case of potability of the devices.

(2) AI-based Chat-Bot Agent: Prevention becomes the highest priority in case of dangerous attacks. It is possibly carried from the study of the users engaged with internet services. AI-based chat-bot agent is governed by Cyber-hygiene services in which users can be warned against the repercussions activities which belongs to highly untrusted resource. Thus, training the users to avoid from cyber-attacks walls is suggested one of the most feasible solution for malware prevention.

(3) Obfuscation and evasion: A ransomware detection system is not stationery and detection is a nonstationary. Therefore, it is very important to cope up Obfuscation methods and evasion of ransomware. It enhances the accuracy of detection system with faster alarming conditions.
(4) Versatile featured dataset: The system must be trained on a highly rich featured dataset which can maintain all possible patterns of dynamic ransomware. detection model may be developed to cope with changing features if multi-featured dataset can be developed as benchmark.
(5) Dynamic population drift: The growth of ransomware follows experiential time which results more more sophisticated its variants with time. Hence, the studies of population drift can provide necessary support to deal the pattern of ransomware.
(6) Restore and Backup: The backedup files of the device can assist to restore new devices after system crashed. Same process in found in all OS refereed as recurring process for backup functionality. The configuration once established supports in mitigating the loss caused by malware.
(7) Pre-Encryption ransomware detection: The limited scope of early detection of ransomware attack does not meet the challenges to face the new variants. Therefore, pre-encryption is utmost important for robust detection.
(8) Limited features for deep network: Deep network models till date suffer from lack of sufficient data for training the model. Therefore, generating rich feature sets is highly demanded.
(9) Novel Elliptic Curve Cryptography (ECC): Elliptic Curve Cryptography (ECC) works based on certificateless remotely random authentication which emphasis to provide secured communication from existing unsecured channels. In this scenario, the adversary never succeed to break the real information of user's identity.
(10) Common Vulnerabilities and Exploits (CVE): Mainly the system pron to attack when it is guaranteed to patch with vulnerabilities and strong cyberhygiene. In general Zero-Day vulnerabilities is given higher privilege than applying malicious actors by the penetration testers. It enables to fed the vulnerabilities into a file which allows CVE publicly. Lack of these vulnerabilities patched by developers leads a strong chain of attacks. Therefore, creating a specific server for taking care of latest CEVs can address the robustness of the system with secured preventive measures.

(11) Windows-based Escalated Services: In general, all the Windows-based devices are highly susceptible to malware attacks. Since the authentication policies in Windows can be abused by malicious easily. The privileged escalation is very common technique to take unauthorized access by malicious execution. User Account Control (UAC) and the mechanism of hijacking DLL are two important ways to bypass the privilege escalation. Despite from the avoidance techniques, power defender and Controlled Power Access (CPA) are strongly advanced techniques like Process Injections, Anti-Analysis mechanisms, and the hooking process of APIs.

6 Conclusion

From the studies of several chronological analysis on security threats aroused from the serious situation like world war among the countries, it is highly voted that protection against cyber attacks and germ-warfare is more significant than physical wars. The recognition of enemy in both kind of the strategic wars remains hidden and more dangerous. In the same scenario, threatening by ransomware attacks increases the probability of wining the war by making the enemy helpless. Therefore, detection and prevention from cyberwar, every nation put at highest attention. As a results the demand of cyber-security analyst is growing exponentially to meet the dangerous threats and making the nation safe and secured.

The major aspects to model the security threats include password authentication, bio-metric traits, and classification based on detecting the nature of attacks. The detection scheme based on classification approach can be promoted by utilizing popularly growing machine learning and deep learning algorithms. In the continuation of image-based malware detection with deep learning, generating sequences of malware codes and corresponding images can assist to construct a real-world model. It can jointly support to deeply leaned network from the malware images and android-based malware detection system.

To generate the sequence of malware code as well as the malware images can be seen an important aspect for developing a robust malware detection. Analyzing the generated code and compare it with the real data is the next challenge for designing a complete system. In current scenario, with the rich feature set of Android, a lot of apps has been installed. This makes an interesting take to analyze various features of Android based malware and further research directions can be extended.

Acknowledgment

This work was supported by Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia, under the internal grant RDU210321.

References

[1] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "Imcfn: Image-based malware classification using fine-tuned convolutional neural network architecture," Computer Networks, vol. 171, p. 107138, 2020.

[2] M. Alazab, M. Alazab, A. Shalaginov, A. Mesleh, and A. Awajan, "Intelligent mobile malware detection using permission requests and api calls," Future Generation Computer Systems, vol. 107, pp. 509–521, 2020.

[3] K. A. Talha, D. I. Alper, and C. Aydin, "Apk auditor: Permission-based android malware detection system," Digital Investigation, vol. 13, pp. 1–14, 2015.

[4] K. Xu, Y. Li, and R. H. Deng, "Iccdetector: Icc-based malware detection on android," IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1252–1264, 2016.

[5] P. Shemitha and J. P. M. Dhas, "Research perceptions on ransomware attack: a complete analysis on conventional authentication protocols in network," Evolutionary Intelligence, pp. 1–16, 2020.

[6] J.-Y. Kim and S.-B. Cho, "Obfuscated malware detection using deep generative model based on global/local features," Computers & Security, vol. 112, p. 102501, 2022.

[7] A. Arora, S. K. Peddoju, and M. Conti, "Permpair: Android malware detection using permission pairs," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1968–1982, 2019.

[8] N. Kumar and N. Sukavanam, "Weakly supervised deep network for spatiotemporal localization and detection of human actions in wild conditions," The Visual Computer, vol. 36, no. 9, pp. 1809–1821, 2020.

[9] N. Kumar, "Large scale deep network architecture of cnn for unconstraint visual activity analytics," in International Conference on Intelligent Systems Design and Applications. Springer, 2017, pp. 251–261.

[10] F. A Ghaleb, F. Saeed, M. Al-Sarem, B. Ali Saleh Al-rimy, W. Boulila, A. Eljialy, K. Aloufi, and M. Alazab, "Misbehavior-aware on-demand

collaborative intrusion detection system using distributed ensemble learning for vanet," Electronics, vol. 9, no. 9, p. 1411, 2020.

[11] B. Geluvaraj, P. Satwik, and T. Ashok Kumar, "The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace," in International Conference on Computer Networks and Communication Technologies. Springer, 2019, pp. 739–747.

[12] S. I. Bae, G. B. Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," Concurrency and Computation: Practice and Experience, vol. 32, no. 18, p. e5422, 2020.

[13] K. Sunitha Krishnan and S. M. Thampi, "Deep learning approaches for iot security in the big data era," in Combating Security Challenges in the Age of Big Data. Springer, 2020, pp. 105–135.

[14] H. Faris, M. Habib, I. Almomani, M. Eshtay, and I. Aljarah, "Optimizing extreme learning machines using chains of salps for efficient android ransomware detection," Applied Sciences, vol. 10, no. 11, p. 3706, 2020.

[15] B. A. S. Al-rimy, M. A. Maarof, Y. A. Prasetyo, S. Z. M. Shaid, and A. F. M. Ariffin, "Zero-day aware decision fusion-based model for crypto-ransomware early detection," International Journal of Integrated Engineering, vol. 10, no. 6, 2018.

[16] N. Andronio, S. Zanero, and F. Maggi, "Heldroid: Dissecting and detecting mobile ransomware," in international symposium on recent advances in intrusion detection. Springer, 2015, pp. 382–404.

[17] F. Mercaldo, V. Nardone, A. Santone, and C. A. Visaggio, "Ransomware steals your phone. formal methods rescue it," in International Conference on Formal Techniques for Distributed Objects, Components, and Systems. Springer, 2016, pp. 212–221.

[18] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, and F. Martinelli, "R-packdroid: Api package-based characterization and detection of mobile ransomware," in Proceedings of the symposium on applied computing, 2017, pp. 1718–1723.

[19] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based onngram of opcodes," Future Generation Computer Systems, vol. 90, pp. 211–221, 2019.

[20] B. Zhang, W. Xiao, X. Xiao, A. K. Sangaiah, W. Zhang, and J. Zhang, "Ransomware classification using patch-based cnn and self-attention

network on embedded n-grams of opcodes," Future Generation Computer Systems, vol. 110, pp. 708–720, 2020.

[21] N. Scaife, H. Carter, P. Traynor, and K. R. Butler, "Cryptolock (and drop it): stopping ransomware attacks on user data," in 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2016, pp. 303–312.

[22] Y. Feng, C. Liu, and B. Liu, "Poster: A new approach to detecting ransomware with deception," in 38th IEEE Symposium on Security and Privacy, 2017.

[23] C. Moore, "Detecting ransomware with honeypot techniques," 2016.

[24] K. Cabaj, P. Gawkowski, K. Grochowski, and D. Osojca, "Network activity analysis of cryptowall ransomware," Przeglad Elektrotechniczny, vol. 91, no. 11, pp. 201–204, 2015.

[25] A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," Journal of Computer Virology and Hacking Techniques, vol. 13, no. 1, pp. 1–12, 2017.

[26] B. Lokuketagoda, M. P. Weerakoon, U. M. Kuruppu, A. N. Senarathne, and K. Y. Abeywardena, "R-killer: An email based ransomware protection tool," in 2018 13th International Conference on Computer Science & Education (ICCSE). IEEE, 2018, pp. 1–7.

[27] S. Kok, A. Abdullah, and N. Jhanjhi, "Early detection of cryptoransomware using pre-encryption detection algorithm," Journal of King Saud University-Computer and Information Sciences, 2020.

[28] A. Ashraf, A. Aziz, U. Zahoora, M. Rajarajan, and A. Khan, "Ransomware analysis using feature engineering and deep neural networks," arXiv preprint arXiv:1910.00286, 2019.

[29] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," Computers & Security, vol. 74, pp. 144–166, 2018.

[30] M. Shukla, S. Mondal, and S. Lodha, "Poster: Locally virtualized environment for mitigating ransomware threat," in proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 1784–1786.

[31] J. A. H. Silva and M. Hernández-Alvarez, "Large scale ransomware detection by cognitive security," in 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). IEEE, 2017, pp. 1–4.

[32] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "A 0-day aware crypto-ransomware early behavioral detection framework," in International Conference of Reliable Information and Communication Technology. Springer, 2017, pp. 758–766.

[33] K. C. Roy and Q. Chen, "Deepran: Attention-based bilstm and crf for ransomware early detection and classification," Information Systems Frontiers, vol. 23, no. 2, pp. 299–315, 2021.

[34] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—a state of the art survey," ACM Computing Surveys (CSUR), vol. 52, no. 5, pp. 1–48, 2019.

[35] G. Jacob, P. M. Comparetti, M. Neugschwandtner, C. Kruegel, and G. Vigna, "A static, packer-agnostic filter to detect similar malware samples," in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, 2012, pp. 102–122.

[36] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." in Ndss, vol. 14, 2014, pp. 23–26.

[37] S. Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik, "A new android malware detection approach using bayesian classification," in 2013 IEEE 27th international conference on advanced information networking and applications (AINA). IEEE, 2013, pp. 121–128.

[38] A. Armando, G. Chiarelli, G. Costa, G. De Maglie, R. Mammoliti, and A. Merlo, "Mobile app security analysis with the maveric static analysis module." J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., vol. 5, no. 4, pp. 103–119, 2014.

[39] Y. Pan, X. Ge, C. Fang, and Y. Fan, "A systematic literature review of android malware detection using static analysis," IEEE Access, vol. 8, pp. 116363–116379, 2020.

[40] V. Syrris and D. Geneiatakis, "On machine learning effectiveness for malware detection in android os using static analysis data," Journal of Information Security and Applications, vol. 59, p. 102794, 2021.

[41] F. Idrees, M. Rajarajan, M. Conti, T. M. Chen, and Y. Rahulamathavan, "Pindroid: A novel android malware detection system using ensemble learning methods," Computers & Security, vol. 68, pp. 36–46, 2017.

[42] A. Feizollah, N. B. Anuar, R. Salleh, G. Suarez-Tangil, and S. Furnell, "Androdialysis: Analysis of android intent effectiveness in malware detection," computers & security, vol. 65, pp. 121–134, 2017.

[43] I. You and K. Yim, "Malware obfuscation techniques: A brief survey," in 2010 International conference on broadband, wireless computing, communication and applications. IEEE, 2010, pp. 297–300.
[44] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, "Paybreak: Defense against cryptographic ransomware," in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, 2017, pp. 599–611.
[45] S. Gupta, H. Sharma, and S. Kaur, "Malware characterization using windows api call sequences," in International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, 2016, pp. 271–280.
[46] Y. Ki, E. Kim, and H. K. Kim, "A novel approach to detect malware based on api call sequence analysis," International Journal of Distributed Sensor Networks, vol. 11, no. 6, p. 659101, 2015.
[47] S. Peisert, M. Bishop, S. Karin, and K. Marzullo, "Analysis of computer intrusions using sequences of function calls," IEEE Transactions on dependable and secure computing, vol. 4, no. 2, pp. 137–150, 2007.
[48] Y. Qiao, Y. Yang, L. Ji, and J. He, "Analyzing malware by abstracting the frequent itemsets in api call sequences," in 2013 12th IEEE international conference on trust, security and privacy in computing and communications. IEEE, 2013, pp. 265–270.
[49] U. Bayer, C. Kruegel, and E. Kirda, TTAnalyze: A tool for analyzing malware. Citeseer, 2006.
[50] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," arXiv preprint arXiv:1609.03020, 2016.
[51] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the gordian knot: A look under the hood of ransomware attacks," in International conference on detection of intrusions and malware, and vulnerability assessment. Springer, 2015, pp. 3–24.
[52] A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barenghi, S. Zanero, and F. Maggi, "Shieldfs: a self-healing, ransomware-aware filesystem," in Proceedings of the 32nd annual conference on computer security applications, 2016, pp. 336–347.
[53] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification," in Proceedings of the sixth ACM conference on data and application security and privacy, 2016, pp. 183–194.

[54] Y. Zhang, Y. Sui, S. Pan, Z. Zheng, B. Ning, I. Tsang, and W. Zhou, "Familial clustering for weakly-labeled android malware using hybrid representation learning," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3401–3414, 2019.

[55] R. Richardson and M. M. North, "Ransomware: Evolution, mitigation and prevention," International Management Review, vol. 13, no. 1, p. 10, 2017.

[56] A. Cimitile, F. Mercaldo, V. Nardone, A. Santone, and C. A. Visaggio, "Talos: no more ransomware victims with formal methods," International Journal of Information Security, vol. 17, no. 6, pp. 719–738, 2018.

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

[58] J. A. H. Silva, L. I. B. López, Á. L. V. Caraguay, and M. HernándezÁlvarez, "A survey on situational awareness of ransomware attacks—detection and prevention parameters," Remote Sensing, vol. 11, no. 10, 2019.

[59] S. Alsoghyer and I. Almomani, "On the effectiveness of application permissions for android ransomware detection," in 2020 6th conference on data science and machine learning applications (CDMA). IEEE, 2020, pp. 94–99.

[60] U. Adamu and I. Awan, "Ransomware prediction using supervised learning algorithms," in 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2019, pp. 57–63.

[61] M. Humayun, N. Jhanjhi, A. Alsayat, and V. Ponnusamy, "Internet of things and ransomware: Evolution, mitigation and prevention," Egyptian Informatics Journal, vol. 22, no. 1, pp. 105–117, 2021.

[62] S. Song, B. Kim, and S. Lee, "The effective ransomware prevention technique using process monitoring on android platform," Mobile Information Systems, vol. 2016, 2016.

[63] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in Proceedings of the 2014 ACM conference on SIGCOMM, 2014, pp. 371–372.

[64] G. Cusack, O. Michel, and E. Keller, "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, 2018, pp. 1–6.

[65] D. Kim and S. Kim, "Design of quantification model for ransom ware prevent," World Journal of Engineering and Technology, vol. 3, no. 03, p. 203, 2015.

[66] J.-Y. Paik, K. Shin, and E.-S. Cho, "Poster: Self-defensible storage devices based on flash memory against ransomware," in Proceedings of IEEE Symposium on Security and Privacy, 2016.

[67] M. Alam, S. Sinha, S. Bhattacharya, S. Dutta, D. Mukhopadhyay, and A. Chattopadhyay, "Rapper: Ransomware prevention via performance counters," arXiv preprint arXiv:2004.01712, 2020.

[68] M. Basnet, S. Poudyal, M. H. Ali, and D. Dasgupta, "Ransomware detection using deep learning in the scada system of electric vehicle charging station," in 2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America). IEEE, 2021, pp. 1–5.

[69] Y. Li, J. Jang, X. Hu, and X. Ou, "Android malware clustering through malicious payload mining," in International symposium on research in attacks, intrusions, and defenses. Springer, 2017, pp. 192–214.

[70] W. Wang, Y. Li, X. Wang, J. Liu, and X. Zhang, "Detecting android malicious apps and categorizing benign apps with ensemble of classifiers," Future generation computer systems, vol. 78, pp. 987–994, 2018.

[71] P. Xu and A. E. Khairi, "Android-coco: Android malware detection with graph neural network for byte-and native-code," arXiv preprint arXiv:2112.10038, 2021.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.

[73] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104–3112.

[74] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," ieee Computational intelligenCe magazine, vol. 13, no. 3, pp. 55–75, 2018.

[75] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, and Y. Xiang, "Code analysis for intelligent cyber systems: A data-driven approach," Information sciences, vol. 524, pp. 46–58, 2020.

[76] N. Kumar, "Recent issues with machine vision applications for deep network architectures," in Cognitive Computing Systems. Apple Academic Press, 2021, pp. 267–284.

[77] R. Feng, S. Chen, X. Xie, G. Meng, S.-W. Lin, and Y. Liu, "A performance-sensitive malware detection system using deep learning on mobile devices," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1563–1578, 2020.

[78] Z. Wang, G. Li, Z. Zhuo, X. Ren, Y. Lin, and J. Gu, "A deep learning method for android application classification using semantic features," Security and Communication Networks, vol. 2022, 2022.

[79] D. Barrera, H. G. Kayacik, P. C. Van Oorschot, and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to android," in Proceedings of the 17th ACM conference on Computer and communications security, 2010, pp. 73–84.

[80] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Android permissions: a perspective combining risks and benefits," in Proceedings of the 17th ACM symposium on Access Control Models and Technologies, 2012, pp. 13–22.

[81] H. Peng, C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Using probabilistic generative models for ranking risks of android apps," in Proceedings of the 2012 ACM conference on Computer and communications security, 2012, pp. 241–252.

[82] Y. Aafer, W. Du, and H. Yin, "Droidapiminer: Mining api-level features for robust malware detection in android," in International conference on security and privacy in communication systems. Springer, 2013, pp. 86–103.

[83] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behaviorbased malware detection system for android," in Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, 2011, pp. 15–26.

[84] S. Chakradeo, B. Reaves, P. Traynor, and W. Enck, "Mast: Triage for market-scale mobile malware analysis," in Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks, 2013, pp. 13–24.

[85] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in 2018 9th IFIP international conference on new technologies, mobility and security (NTMS). IEEE, 2018, pp. 1–5.

[86] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," Journal of Computer Virology and Hacking Techniques, vol. 15, no. 1, pp. 15–28, 2019.

[87] N. Marastoni, R. Giacobazzi, and M. Dalla Preda, "Data augmentation and transfer learning to classify malware images in a deep learning context," Journal of Computer Virology and Hacking Techniques, vol. 17, no. 4, pp. 279–297, 2021.

[88] N. Bhodia, P. Prajapati, F. Di Troia, and M. Stamp, "Transfer learning for image-based malware classification," arXiv preprint arXiv:1903.11551, 2019.

[89] P. Prajapati and M. Stamp, "An empirical analysis of image-based learning techniques for malware classification," in Malware Analysis Using Artificial Intelligence and Deep Learning. Springer, 2021, pp. 411–435.

[90] X. Pei, X. Deng, S. Tian, L. Zhang, and K. Xue, "A knowledge transfer-based semi-supervised federated learning for iot malware detection," IEEE Transactions on Dependable and Secure Computing, 2022.

[91] S. Yajamanam, V. R. S. Selvin, F. Di Troia, and M. Stamp, "Deep learning versus gist descriptors for image-based malware classification." in Icissp, 2018, pp. 553–561.

[92] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in Proceedings of the ACM International Conference on Image and Video Retrieval, 2009, pp. 1–8.

[93] D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, "Image-based malware classification using ensemble of cnn architectures (imcec)," Computers & Security, vol. 92, p. 101748, 2020.

[94] M. Jain, W. Andreopoulos, and M. Stamp, "Convolutional neural networks and extreme learning machines for malware classification," Journal of Computer Virology and Hacking Techniques, vol. 16, no. 3, pp. 229–244, 2020.

[95] Y. A. Ahmed, B. Koçer, S. Huda, B. A. S. Al-rimy, and M. M. Hassan, "A system call refinement-based enhanced minimum redundancy maximum relevance method for ransomware early detection," Journal of Network and Computer Applications, vol. 167, p. 102753, 2020.

[96] H. Zuhair and A. Selamat, "Rands: A machine learning-based anti-ransomware tool for windows platforms," in Advancing Technology

Industrialization Through Intelligent Software Methodologies, Tools and Techniques. IOS Press, 2019, pp. 573–587.

[97] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K.-K. R. Choo, and D. E. Newton, "Drthis: Deep ransomware threat hunting and intelligence system at the fog layer," Future Generation Computer Systems, vol. 90, pp. 94–104, 2019.

[98] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection," Future Generation Computer Systems, vol. 101, pp. 476–491, 2019.

[99] S. Kok, A. Azween, and N. Jhanjhi, "Evaluation metric for cryptoransomware detection using machine learning," Journal of Information Security and Applications, vol. 55, p. 102646, 2020.

[100] L. Fernandez Maimo, A. Huertas Celdran, A. L. Perales Gomez, F. J. Garcia Clemente, J. Weimer, and I. Lee, "Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments," Sensors, vol. 19, no. 5, p. 1114, 2019.

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

[102] S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, "On the classification of microsoft-windows ransomware using hardware profile," PeerJ Computer Science, vol. 7, p. e361, 2021.

[103] B. Jethva, I. Traoré, A. Ghaleb, K. Ganame, and S. Ahmed, "Multi-layer ransomware detection using grouped registry key operations, file entropy and file signature monitoring," Journal of Computer Security, vol. 28, no. 3, pp. 337–373, 2020.

[104] S. K. Shaukat and V. J. Ribeiro, "Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning," in 2018 10th International Conference on Communication Systems & Networks (COMSNETS). IEEE, 2018, pp. 356–363.

[105] S. Aurangzeb, M. Aleem, M. A. Iqbal, M. A. Islam et al., "Ransomware: a survey and trends," J. Inf. Assur. Secur, vol. 6, no. 2, pp. 48–58, 2017.

[106] N. K. Popli and A. Girdhar, "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, 2019, pp. 65–80.

[107] B. A. S. Al-Rimy, M. A. Maarof, M. Alazab, F. Alsolami, S. Z. M. Shaid, F. A. Ghaleb, T. Al-Hadhrami, and A. M. Ali, "A pseudo feedbackbased annotated tf-idf technique for dynamic crypto-ransomware preencryption boundary delineation and features extraction," IEEE Access, vol. 8, pp. 140586–140598, 2020.

[108] R. Vinayakumar, K. Soman, K. S. Velan, and S. Ganorkar, "Evaluating shallow and deep networks for ransomware detection and classification," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017, pp. 259–265.

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

[110] M. M. Hasan and M. M. Rahman, "Ranshunt: A support vector machines based ransomware analysis framework with integrated feature set," in 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE, 2017, pp. 1–7.

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

[112] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Ransomware, threat and detection techniques: A review," Int. J. Comput. Sci. Netw. Secur, vol. 19, no. 2, p. 136, 2019.

[113] C. Do Xuan and D. Huong, "A new approach for apt malware detection based on deep graph network for endpoint systems," Applied Intelligence, pp. 1–20, 2022.

[114] J. Hwang, J. Kim, S. Lee, and K. Kim, "Two-stage ransomware detection using dynamic analysis and machine learning techniques," Wireless Personal Communications, vol. 112, no. 4, pp. 2597–2609, 2020.

[115] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "{UNVEIL}: A {Large-Scale}, automated approach to detecting ransomware," in 25th USENIX security symposium (USENIX security 16), 2016, pp. 757–772.

[116] A. Mallik, A. Khetarpal, and S. Kumar, "Conrec: malware classification using convolutional recurrence," Journal of Computer Virology and Hacking Techniques, pp. 1–17, 2022.

[117] J. Zhu, J. Jang-Jaccard, A. Singh, I. Welch, A.-S. Harith, and S. Camtepe, "A few-shot meta-learning based siamese neural network using entropy features for ransomware classification," Computers & Security, vol. 117, p. 102691, 2022.

[118] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "Mitigating adversarial evasion attacks of ransomware using ensemble learning," Computers and Electrical Engineering, vol. 100, p. 107903, 2022.

[119] F. Manavi and A. Hamzeh, "A novel approach for ransomware detection based on pe header using graph embedding," Journal of Computer Virology and Hacking Techniques, pp. 1–12, 2022.

[120] S. I. Imtiaz, S. ur Rehman, A. R. Javed, Z. Jalil, X. Liu, and W. S. Alnumay, "Deepamd: Detection and identification of android malware using high-efficient deep artificial neural network," Future Generation computer systems, vol. 115, pp. 844–856, 2021.

[121] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in 2015 10th international conference on malicious and unwanted software (MALWARE). IEEE, 2015, pp. 11–20.

[122] Z. Liu, R. Wang, N. Japkowicz, D. Tang, W. Zhang, and J. Zhao, "Research on unsupervised feature learning for android malware detection.

[123] Y. Ye, L. Chen, S. Hou, W. Hardy, and X. Li, "Deepam: a heterogeneous deep learning framework for intelligent malware detection," Knowledge and Information Systems, vol. 54, no. 2, pp. 265–285, 2018.

[124] F. Naït-Abdesselam, A. Darwaish, and C. Titouna, "Malware forensics: Legacy solutions, recent advances, and future challenges," in Advances in Computing, Informatics, Networking and Cybersecurity. Springer, 2022, pp. 685–710.

[125] A. Pekta¸s and T. Acarman, "Classification of malware families based on runtime behaviors," Journal of information security and applications, vol. 37, pp. 91–100, 2017.

[126] D. Yuxin and Z. Siyi, "Malware detection based on deep learning algorithm," Neural Computing and Applications, vol. 31, no. 2, pp. 461–472, 2019.

[127] S. Garg and N. Baliyan, "M2vmapper: Malware-to-vulnerability mapping for android using text processing," Expert Systems with Applications, vol. 191, p. 116360, 2022.

[128] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[129] X. Xiao, S. Zhang, F. Mercaldo, G. Hu, and A. K. Sangaiah, "Android malware detection based on system call sequences and lstm," Multimedia Tools and Applications, vol. 78, no. 4, pp. 3979–3999, 2019.
[130] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," Information Sciences, vol. 460, pp. 83–102, 2018.
[131] M. Q. Li, B. C. Fung, P. Charland, and S. H. Ding, "I-mad: Interpretable malware detector using galaxy transformer," Computers & Security, vol. 108, p. 102371, 2021.
[132] J. A. Gómez-Hernández, L. Álvarez-González, and P. García-Teodoro, "R-locker: Thwarting ransomware action through a honeyfile-based approach," Computers & Security, vol. 73, pp. 389–398, 2018.
[133] H. Alshahrani, H. Mansourt, S. Thorn, A. Alshehri, A. Alzahrani, and H. Fu, "Ddefender: Android application threat detection using static and dynamic analysis," in 2018 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2018, pp. 1–6.
[134] T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," computers & security, vol. 73, pp. 137–155, 2018.
[135] A. Kumar, K. Kuppusamy, and G. Aghila, "A learning model to detect maliciousness of portable executable using integrated feature set," Journal of King Saud University-Computer and Information Sciences, vol. 31, no. 2, pp. 252–265, 2019.
[136] J. Stiborek, T. Pevny, and M. Rehák, "Multiple instance learning for ' malware classification," Expert Systems with Applications, vol. 93, pp. 346–357, 2018.
[137] C.-H. Lin, H.-K. Pao, and J.-W. Liao, "Efficient dynamic malware analysis using virtual time control mechanics," Computers & Security, vol. 73, pp. 359–373, 2018.
[138] S. Chen, M. Xue, L. Fan, S. Hao, L. Xu, H. Zhu, and B. Li, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," computers & security, vol. 73, pp. 326–344, 2018.
[139] N. Hampton, Z. Baig, and S. Zeadally, "Ransomware behavioural analysis on windows platforms," Journal of information security and applications, vol. 40, pp. 44–51, 2018.

[140] J. Stiborek, T. Pevny, and M. Rehák, "Probabilistic analysis of dynamic' malware traces," Computers & Security, vol. 74, pp. 221–239, 2018.

[141] S. Hou, A. Saas, L. Chen, and Y. Ye, "Deep4maldroid: A deep learning framework for android malware detection based on linux kernel system call graphs," in 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW). IEEE, 2016, pp. 104–111.

[142] D. W. Fernando and N. Komninos, "Fesa: Feature selection architecture for ransomware detection under concept drift," Computers & Security, vol. 116, p. 102659, 2022.

Biographies

Syed Shuja Hussain has received BS degree in Computer Engineering from the Sir Syed University of Engineering and Technology (SSUET), Pakistan and MS degree in Telecommunication Engineering from University of Engineering and Technology (UET) Peshawar, Pakistan. He is pursuing a Ph.D. from the Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia. He has been involved in research work on Android malware analysis.

Mohd Faizal Ab Razak has distinctively received his PhD from University of Malaya and Master of Computer Science (Networking) from Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia. He is currently a lecturer and researcher at Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia. His area of research includes Mobile Computing, Intrusion Detection System, risk assessment, network security and Mobile Security.

Ahmad Firdaus distinctively received his PhD from University of Malaya (UM), Malaysia. He also obtained his Master of Computer Science (Networking) from Universiti Teknologi Mara (UiTM), Malaysia. He is currently a senior lecturer at the Faculty of Computing at Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia. His area of research includes Mobile Security, Intrusion Detection System and Blockchain.