RES MILITARIS
Social Science Journal

ADVANCES IN MALWARE DETECTION APPROACHES USING MACHINE AND DEEP LEARNING

Pushpendra Dwivedi, C. S. Raghuvanshi, Hari Om Sharan

Faculty of Engineering and Technology, Rama University Kanpur, Uttar Pradesh
pushpendradwivedi10@gmal.com

***Abstract*—** Many institutions have suffered significant financial losses as a result of malware's rapid growth over the previous decade, according to studies. Anti-malware businesses have come up with a variety of ways to protect against these threats. The anti-malware profession is facing new problems due to the increasing speed, size, and complexities of malware. During malware detection, malware classification is a key element of malware analysis. In order to determine whether a given sample is infected with malware or not, a range of analysis methods can be utilized, including static analysis, dynamic analysis, and hybrid analysis techniques. After examination, virus and benign files may be easily distinguished by their distinct properties. Detection systems are more successful when they can identify specific malware traits using analytical approaches. Static and dynamic analysis tools may be used to build up analysis settings in a variety of ways. The malware classifiers are trained in the second step. Malware categorization used to be done using conventional techniques, however nowadays machine learning algorithms are utilized since they are able to handle the increasing complexity and speed of malware evolution. Machine and deep learning approaches have advanced the field of malware detection by enabling the development of more effective and efficient techniques. This research paper provides a comprehensive examination of the current state-of-the-art in malware detection techniques, focusing specifically on the latest advancements and approaches utilizing machine learning and deep learning methods.

Keywords—Malware classification, anomaly detection, behavior analysis, Hybrid analysis, Machine Learning, Deep Learning, Convolutional Neural Networks.

I. INTRODUCTION

Programs that are designed to target and harm digital devices (windows, android, scada, cloud systems) are known as "malware". Based on the way they behave and the activities they execute, malicious software may be divided into a wide variety of types, such as trojans, rootkit, and virus attacks [1]. The ultimate goal of cybercriminals is to gain financial gain, obtain confidential and sensitive data, exploit computing resources, and disrupt network services, among other malicious activities, by targeting vulnerable computer systems. [2][3].

One of the most serious threats to network security is malicious software. Malware is classified based on how it influences the computer, the program's functioning, and the method for its growth [4]. after each iteration, the structure of malware evolves. This is a fundamental difference between the two approaches. The malware's dynamic nature makes it more difficult to identify and quarantine [5]. Malware detection relies heavily on signatures, heuristics, normalization, and machine and deep learning approach. A large feature library may be compiled using traditional signature-based approaches by extracting binary signatures from malware, but this takes a long time and effort [6]. In addition, advanced malware like polymorphic, metamorphic, and packed malware, which are extremely difficult to spot and analyse, are created by employing encryption and encoding methods [7].

In recent years, the field of malware detection has witnessed a proliferation of innovative techniques and methodologies, such as multi-signature detection, static analysis, dynamic detection, and heuristic detection, among others, which aim to enhance the accuracy and efficacy of malware detection and prevention. Anti-malware detection technology, on the other hand, is improving all the time. Object-code obfuscation, code restructuring, and other techniques have all been used by malware to adapt their functionality [8]. More than half of the new malware that is created each minute is a variation of an existing virus, according to research by Symantec and McAfee.

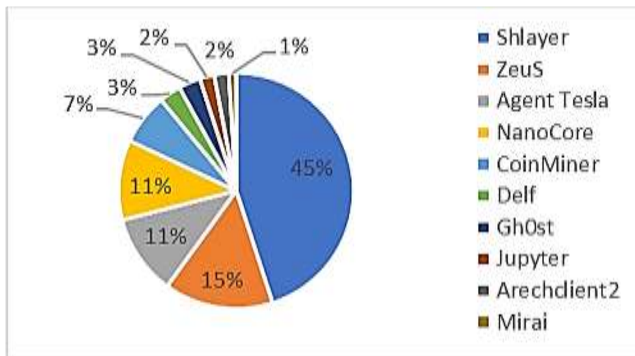


Fig. 1. Top 10 malware in march-24

Machine learning has long been seen as a promising strategy by anti-malware developers. When compared with conventional malware that was wide and accessible, modern malware is more specialized, stealth and has a long-term presence compared to conventional malware that was only executed once [9]. The identification of zero-day infection is difficult since it utilizes newer vulnerabilities that have not yet been disclosed [10]. A wide range of computer science fields now use Artificial Intelligence, ML, and deep learning methodologies, from NLP to malware detection strategies [11]. There are two types of malware characteristics that may be used to train classifiers: dynamic and static malware analysis methodologies.

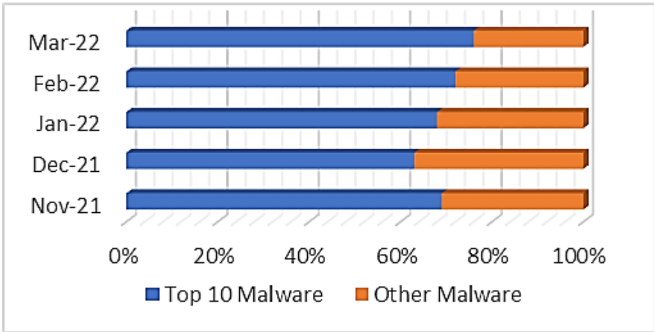


Fig. 2. Contribution of Top 10 malware in from nov-23 to march 2024

To avoid the risk of malware infection, the investigation of malware instances in this study does not involve executing the malware. Instead, static characteristics, including hash values, N-grams, opcodes, string patterns, and portable executable (PE) header information, are extracted and analysed to obtain relevant features for malware detection. [12]. Detection application such as intrusion detection system and anti-malware is built with these features in mind [13]. To avoid infecting the host OS by executing the malicious file, malware instances are run in a restricted virtual environment rather than on the host computer [14].

The paper provides a thorough and detailed examination of the various methods, tools, and approaches used in malware analysis. The second section outlines the previous research conducted in this field, while the third section focuses on malware analysis techniques. The fourth section highlights malware detection approaches, and sections five and six delve into the recent developments and advancements in machine learning and deep learning techniques. Finally, the paper concludes with section seven, summarizing the findings of the study.

II. Related Work

The malware detecting framework created by [12] makes use of adequate API calls. To obtain API calls from the malicious code, a disassembler tool like IDA Pro was utilized in this study.. Using unpacking tools prior to disassembly, malware was analysed. Statically analysing both Windows system32 and malware files yields APIs for both classes. With the help of the retrieved API calls, the support vector machine classifier was trained. The author in [31] suggested a method that recorded malware's API calls, particular addresses, and routines as they interacted with system services in the computer.

Malware targeting Android devices has been studied by author [33]; a multi-feature consensus-based decision fusion adaptive identification component is now being developed to use this malware (MCDF). Static analysis combined with machine learning approaches was used by Srndic et al. [26] to categorise malware samples in their research. Two distinct file formats were examined in this study. Malware developers are now using PDF and SWF files to include executable programmes that impair computer resources. 40,000 SWFs and 440,000 PDFs were examined in this study. The framework offered by this technology allowed the detection of dangerous code contained in PDF and SWF files. The automated methodology for detecting variations of malware classes was proposed by the research work in [4]. The EHNFC classifier suggested by study in [39] uses consent-based components to evolve as a hybrid neuro-fuzzy classifier for Android malware classification. The researcher in [34] built a malware classifier that was able to handle polymorphic malware by portraying them in the form of pictures that could catch subtle changes while maintaining the overall architecture.

The study in [36] offered a deep learning strategy to combine static and dynamic analysis for Android applications. The findings suggest that deep learning is a promising technique for detecting Android malware, particularly in situations when more preliminary data is readily available.. In comparison to typical machine learning approaches, DroidDetector is able to achieve 96.76 percent detection performance and accuracy. A probabilistic Neural Network (PNN) was suggested by the researchers in [41] for identifying malicious activity in network data. Protocols, and jitters were employed as feature vectors in the study. Using a combination of the Particle Swarm Optimization (PSO) and PNN algorithms, the suggested method was then put into practice and shown to be 96.5 percent effective in detecting fraudulent network traffic.

The research work in [52] Proposed ScaleMalNet for zero-day malware, a deep learning framework based on static, dynamic, and image processing based for malware identification. Galal et al. (2016) [35] offered a behavior-based characteristics approach to defining what constitutes malware. In order to remove the proposed model, they first perform dynamic inspection on a typically late malware dataset in a controlled virtual environment, where they collect traces of API calls produced by malware samples. Higher-level features, or "actions," are generated from the traces. Malicious traffic classified by Arivudainambi, Varun, et al. [40] using a network traffic analysis methodology. Improved anti-network traffic methodological approaches necessitated the use of PCA. The suggested strategy was put into practice by executing 1,000 malicious files in the sandboxes Noriben, Cuckoo, and Limon, among others. This method yielded a 99 percent success rate in detecting malware.

III. Malware Analysis Techniques

Analysis of malware samples is performed to identify the properties that may be applied to determine them. Since malware is becoming more sophisticated in the lifecycle, knowledge about cryptic malware protection has emerged as a critical issue in malware detection, according to machine learning methodologies [15]. Additionally, there still are two types of malware analysis that are often used in the process of identifying malicious applications [16]. malware detection techniques based on ML strategies uses feature extraction for the analysis. These features (API calls, Assembly, and Binary) used machine learning methodologies for classifying malware.

A. Static Analysis Techniques

Features extraction and classifying are the first two stages of malware detection. ML algorithms may use malware features as input. A feature is considered static if it can be retrieved without the use of malware. We study malicious files, either even without the utilizing reverse engineering approach on the malicious sample under considerationThe structure of malware may be determined by disassembling executable harmful files into assembly language code and then analyzing the resulting code. This is accomplished with the use of popular disassemblers and debuggers as Ollydbg, IDA Pro, capstone, and WinDbg [17]. Binary files can be disassembled into assembly language with the help of these applications.

B. Dynamic Analysis Techniques

The classification and analysis of malware, malicious files are executed in a controlled environment at their runtime activities are monitored [18]. Virtualization software, such as Virtual Box or VMware, is used to build virtual environments on the cloud. When a suspected program or file is executed in a controlled environment, a variety of activities can be discovered, including the generation of different files, the identification and monitoring of system or user files, the addition of new entries, the modification of registry keys, the accessing of URLs, the use of API calls, the downloading of malware, and the transmission of data to a command-and-control system [19]. The file is classified as either a safe file or a malicious file based on the actions it does. Through the use of dynamic analysis, it is possible to study files that have not been thoroughly disassembled or investigated using the static analysis method.

TABLE I. TOOLS USED FOR STATIC AND DYNAMIC ANALYSIS

Static Analysis Tools	Dynamic Analysis Tools
IDA Pro (dissembler)	ProcMon (logs lve system activity)
Ghidra (dissembler)	PeStudio (Windows executable analyzer)
PeView (PE header information)	Process Hacker (Gathering information of process)
UPX	Wireshark (packet analysis tool)
YARA (string matching)	TCPdump (TCP/IP packet analyser)
x64dbg (reverse engineering)	Regshot (snapshot of registry related files)
HxD	VmWare/VitualBox (virtual machine)
PE-bear	Comodo IMA (malware analysis sandbox)
PeStudio (analysing executables)	Cuckoo Sandbox
IOCFinder	RegMon (registry monitoring)

IV. MALWARE DETECTION APPRAOCHES

The goal of malware detection technologies is to identify and protect computer systems and networks from harmful programs. A variety of input representations are used to identify and categorize malware samples. The infected file's real behaviour must be known in order to identify the malicious file. you.

A. Signature Based Malware Detection Approach

An anti-virus, or malware detection system relies heavily on the use of signatures to identify suspicious activity. This approach works by searching a vast dataset of signatures for specific patterns of viruses. The signature-based method searches for disruptions by referring to a previously specified list of known attacks. Regardless of the fact that this configuration is capable of identifying malware in a wide variety of applications, it needs the regular updating of the specified signature database to maintain its effectiveness. As a result, it is less successful in detecting harmful workouts when using the signature-based method, owing to the constantly evolving nature of versatile malware [20]. Metaheuristic approaches are adopted by the anti-virus provider which can effectively identify the malicious codes to manage their signature [21].

Feature extraction tools like PeView, PeExplorer, PsStudio, HashGenerator are for static feature extraction. Static analysis at code level is achieved using disassembler tools for example Lida, Cpstone, IDA Pro. Malware static features like N-gram [22](n-gram 3: 'mail', 'ili, 'ftw'), String [22]('APIcallname', 'mytime', 'kernal32'), Opcode [23]('ADD', 'SUB', 'MOV, 'PUSH'), Hash Values ('e5dadf6524624f79c3127e247f04b548'), PE Header information [24]('field value', 'checksum', 'size', 'symbol') are extracted for analysis. The challenge of signature-based identification may be reduced to a simple one of string matching. Basically, this implies that it continues looking for a pattern or substring in a huge string dataset. Almost all of the computing time is spent to just this procedure (45 percent to 75 percent of the time) [25]. Aho-Corasick and Boyer-Moore are two of the most often used algorithms for string matching. Deobfuscating every piece of malware is quite difficult, despite the fact that many unpacking techniques are present.

WU Bin et al. (2015) [27]proposed a malware detection model for the mobile phone based on artificial immune based system. As well as varying detectors, a clone and mutation method is applied to increase the detection accuracy. Token-based resemblance and character-based resemblance were combined to create a new similarity matrix, and it was also shown that existing characteristics are specific examples of fuzzy token similarity. Jiannan Wang et al. (2011) [28]developed a signature-based system to solve the problem of fuzzy-token similarity joins. In comparison to other existing signature techniques, it is found that the token-sensitive approach is better. Edit similarity was included as an extension to current signature systems for edit distance.

B. Behaviour Based Malware Detection Approach

The identification of malware in the behaviour-based method is done based on the destructive actions that malware does while it is running. APIs, browser events, and other characteristics are used to specify how the application behaves. Sandboxed environments are required for behaviour-based techniques to work, and the results of the run-time activities are recorded[29]. Virtualization and simulating circumstances are used to run malware and eliminate its behaviours in dynamic systems. An improved comprehension of malware creation and distribution can be attained by a behavior-based approach [16]. This is because malicious files often share their malicious code with one another. Because of this, malware detection systems are being trained to recognize new malware or variations of old malware based on their shared behaviors. Malware that is difficult to identify is another potential target for a behaviour-

Identify applicable funding agency here. If none, delete this text box.

based approach, which is another plus [30]. Malware authors' obfuscation techniques can circumvent signature-based detection systems.. Anomaly refers to a malfunction caused by malicious files and is taught into the behaviour-based approaches in two ways. Malicious files are those that display anomalous behaviour that is inconsistent with the stored behaviour of normal files.

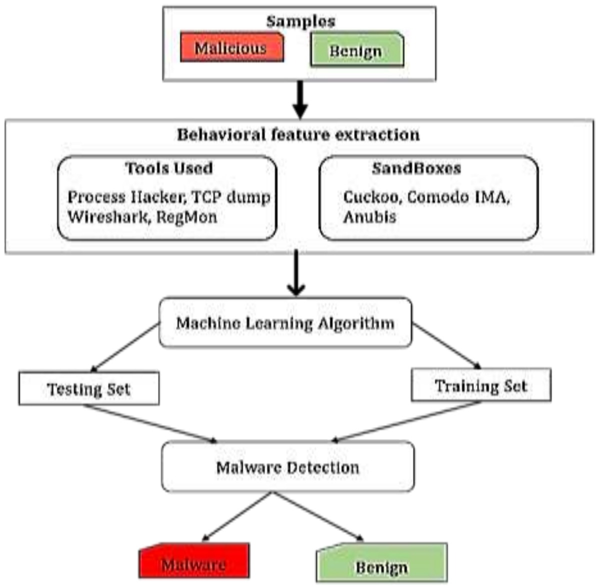


Fig. 3. Behavior based malware detection architecture

Behavioural-based malware detections approaches are discussed in detail in this section. The advanced methods are brought up to identify malware, Bailey et al. (2007) [31] suggested a method that recorded malware's API calls. A new hybrid method, HDM-Analyzer, was proposed by Eskandari et al. (2013) [32], taking into account both dynamic and static inquiry points of interest, while keeping precision at a reasonable level. Because of this, HDM-Analyzer can forecast that most of the fundamental leadership is based on real data, and so has little performance degradation. Sheen et al.(2015) [33] developed MCDF. Malicious record characteristics like the consent-based features and API call-based features are evaluated in order to provide a better discovery by merging the classifiers' choices using the collective method based on the probability hypothesis, which is used to construct a group of classifiers.

Utilizing supervised machine learning techniques, Narayanan et al. (2016) [34] built a malware classifier that was able to handle polymorphic.

Ming et al. (2017)[37] have developed a substitution attack that affects behaviour-based requirements to cover similar behaviours. The main attack approach is to replace a graph of system call dependency with its semantically equivalent variants so that the comparable malware test's secret unique family becomes characteristically distinctive. Malware researchers should thus devote more time and effort to the re-examination of identical samples that may have recently been studied, as a result of this.

It has been suggested by Nikolopoulos et. al (2017) [38] that an unidentified application specimen may be classified as harmful or non-malicious using a graph-based model concerning the relationship between collections of system calls and known malware families.

TABLE II. ANALYSIS OF THE SIGNATURE-BASED MALWARE DETECTION APPROACHED

Author	Data set	Total Sample Set	Feature	Accuracy (%)
Gavrilut et al., 2009 [66]	VX Heaven	29254 (malicious:12817, Benign: 16437)	API Call	88.8
Veeramani and Rai, 2012 [12]	VX Heaven	514 (malicious:214, Benign:300)	API Call	97
Santos et al., 2013 [67]	VX Heaven	2000	Op Code	92.9
Zane Markel and Michael Bilzor, 2014 [68]	Open Malware	164,802 (122799 malicious, 42,003 benign	PE32	97
Fraley et al., 2016 [69]	ClamAV, VirusTotal, VirusShare	3,637 (800 malicious, 2400 benign	API, Weka	99
Boujnouni et al., 2016 [70]	VX Heaven	1258 (658 malicious, 600 benign	N-gram	97
Narra et al., 2016 [71]	VirusShare	7800 malicious	Op Code	98
Kim et al., 2016 [72]	VX Heaven	280868(271095 malicious, 9773 benign)	PE header	99
Chowdhury et al., 2018 [73]	Download.com	52,185 (41,265 malicious, 10,920 benign)	BAM	98.6
Nagano et al. [74]	CCC Dataset	3600(1800 malicious, 1800 benign)	Assembly, DLL, hexdump	99
Lee et al., 2018 [53]	Kaggle	10708 malicious	CNN	98.8
Abiola and Marhusin, 2018 [75]	Open Malware	213,699 malicious, 152,421 benign	N-gram	99

TABLE III. ANALYSIS OF THE BEHAVIOR-BASED MALWARE ANALYSIS

Author	Data set	Total Sample Set	Feature	Accuracy (%)
A.Elhadi et al., 2014 [60]	VX Heaven	514 (Malicious: 416, benign: 98)	API call graph	98
Pirscoveanu et al., 2015 [46]	VirusTotal	42000 samples	API call	98
Narayanan et al., 2016 [34]	Kaggle	10868 samples	images	96.7
Boukhtouta et al., 2016 [61]	Third Party	14400 samples	J48	99
W¨uchner et al., 2017 [62]	Zeus, SpyEye	7507 (malicious:6994, benign:513)	QDFG	96
Nikolopoulos et. al., 2017 [38]	App	2631 malicious, 35 benign	Gr graph	94.7
Ye et al., 2018 [54]	Comodo CSC	20000 (9000 malicious, 9000 benign)	API call	98
Stiborek et al., 2018 [50]	AMP ThreatGrid	143684 malicious, 86707 benign	Dynamic	95.4
Arivudainambi et al., 2019 [40]	internet	1000 malicious samples	NN-PCA	99
Jahromi et al., 2020 [57]	VX Heaven	18830 malicious	Op codes	99.7
Yücel et al., 2020 [63]	ViruSign	123(121 malicious, 6 benign)	Images	99.5
Lashkari et al., 2021 [64]	Memory dump	1900 (1127 legitimate and 773 malware)	Volatile memory	93
Sharma et. al., 2022 [65]	AndroZoo	4300 sample APKs	Reverse Engineering	98.08

V. MACHINE LEARNING FOR MALWARE ANALYSIS AND DETECTION

The categorization and grouping of malwares are greatly aided by Machine Learning (ML). Classifying files as benign or malicious has been the subject of many studies in the literature [42]. Algorithms for machine learning (ML) take into account more characteristics of malicious and benign samples, allowing for the creation of more precise models [43]. As a result of PE's connection to the network, a wealth of information regarding malware is revealed. When malware like a key logger infects a computer, it collects sensitive data about the user and delivers it to the attacker across the network. Neural networks' effectiveness in malware detection is being reproduced in machine learning for information security. It is possible to minimize the dataset's dimensionality in order to save money on training since the machine learning approach takes longer to process a dataset with more data characteristics. Some methods may be used to choose just the most relevant and discriminatory virus features[44]. Training expenses may be reduced as a result. The development of hybrid malware classifiers may address the second adversarial ML difficulty. Call graphs, API calls, and strings are used as features for malware detection, and classifiers such as Support Vector Machine (SVM) [26], Naive Bayes (NB) [45], Random Forest (RF) [46], Decision Tree (DT) [47], Artificial Neural Network (ANN), k-Nearest Neighbours (k-NN) [48], Logistic Regression (LR) [49], and ensemble algorithms like Random Forest (RF) [50], Adaptive Boosting (AB) are used for training [51].

VI. DEEP LEARNING FOR MALWARE CLASSIFICATION AND DETECTION

Machine learning, of which deep learning is a subset, encompasses a wide range of techniques. Unstructured or even unlabelled data may be used to train it. Because it resembles the functioning of the human brain, it gathers information, analyses it, and draws conclusions based on the patterns it discovers about itself. Neurons are the basis of deep learning [52]. In a neural network, each layer is linked to the previous one. Input, output, and a hidden layer are only a few of the many layers that make up a neural network. Input and output are separated by a layer called a hidden layer. The network is described as "deep" if it has more than two levels. Deep learning (DL) is an AI submodule that draws inspiration from how the human brain operates. For deep learning architectures, understanding the meaning of vast volumes of data is a key strength, as well as a capacity to dynamically adapt the derived meaning with fresh data without the requirement for domain expertise. Deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are often used in real-world situations. To train deep learning classifiers, rather than using task-specific algorithms, we use feature learning instead of task-specific algorithms

From a data and feature viewpoint, the examination of the numerous uses of deep learning in malware categorization is done on the basis of static code features and executed code features [53]. Ye et al. (2017) [54] proposed an AutoEncoders built on top of multilayer restricted Boltzmann machines and a layer of associative memory for identifying newly discovered malware. Rathore et al. (2018) [55] find that Random Forest outperformed Deep Neural Network (DNN) using opcode frequency as a feature vector and unsupervised learning in addition to supervised learning for the classification of malware is used.

Ashik et al. (2021) [56] used four data sets applying ML and Deep learning approaches like SVM, NB, J48, and RF. Fine-tuned DNN results were 99.1% and 98.4% on datasets 2, and 3 accordingly.

TABLE IV. ANALYSIS OF THE VARIOUS MACHINE LEARNING ALGORITHM FOR MALWARE DETECTION

Author	Data Set	Features	ML Algorithm							Accuracy (%)
			NB	*KNN*	*SVM*	*RF*	*DT*	*LR*	*ADA*	
Shabtai et al., 2012 [45]	5,677 malware, 20,416 benign	N gram, Op code	Y	Y	Y	Y	Y	Y	Y	RF-95.3, DT-93.7, LR-92.9, SVM-92.1, NB-85.5
Bai et al., 2014 [47]	10521 malicious, 8592 benign	PE files	-	-	-	Y	Y	-	-	RF-98.9
Pirscoveanu et al., 2015 [46]	80,000 malwares	API call				Y				RF-98
Šrndi´c et al. 2016 [26]	32,567 malicious, 407,037 benign	API call, PSI	-	-	Y	-	-	-	-	SVM-95
Shamsul Huda et al., 2017 [58]	967 malicious	API call, PSI	Y	-	Y	Y	Y	-	-	SVM-95.7, RF-94.8, NB-87.7
Jan Stiborek et al., 2018 [50]	143684 malware, 86707 benign	Run time	-	-	Y	Y	-	-	-	SVM-95.4
Shamsul Huda et al., 2018[58]	2000 malicious, 1500 benign	Run time	-	-	Y	-	-	-	-	SVM-98.2
Ghafir et al., 2018 [76]	Network	Run time	-	Y	Y	-	Y	-	Y	Simple DT-84.4, Linear SVM-84.8Medium KNN-80
Singh et al., 2020 [48]	8634 Malicious, 6434 benign	API calls	Y	Y	Y	Y	Y	-	-	RF-99.1, KNN-98.4, SVM-98.1, DT-98.1, NB-85.4
Hemalatha et al. 2020 [49]	7268 Malimg, 8338 BIG2015, 9958 MaleVis	Binary image	Y	Y	Y	Y	Y	Y	Y	RF-82.1, SVM-80, DT-77.7, KNN-76.7, ADA-75.3, LR-56.3, NB-46.9
Elayan and Ahmed, 2021 [77]	347 - Benign 365 - Malware	API call	Y	Y	Y	Y	Y	-	-	Naive Bayes – 93.9, RF – 97.8, DT – 96.6 KNN – 97.2, SVM – 96.2

TABLE V. ANALYSIS OF DEEP LEARNING APPROACH FOR MALWARE DETECTION

Author	Data set/ Classes	Total Sample Set	Feature	Accuracy (%)
Pascanu et al., (2015) [78]	company's database/2	2,50,000	API calls	98.3
Saxe and Berlin, (2016) [79]	Invincea/3	350,016 malicious, 81,910 benign	PE, binaries	95
Tobiyama et al., (2016) [80]	NTT/2	81 malicious process log, 69 benign	CNN, RNN	96
Makandar and Patrot (2016) [81]	Mahenhur	3131 samples	Binary	96.3
Chowdhury et al., (2017) [73]	KDD99, NSL-KDD/	4,94,021	CNN	96
Kabanga and Kim, (2017) [82]	MalIMG/25	9458 images	CNN	98
Kalash et al., (2018) [83]	MalIMG/25	10868 malicious	CNN	98.5
Gibert et al., (2019) [84]	MalIMG/25	9342	Images	98.4
Vasan et al. (2020) [85]	MalIMG /25	9435	CNN	98.8
Hemalatha et al. (2021) [49]	MaleVis, Malicia/2	14,226, 9670 samples	Binary Images	89.4
Darem et al. (2021) [86]	BIG2015/9	4358 odd images	Binary images, XGBoost	99.1

VII. CONCLUSION

The ever-increasing prevalence of malicious software has made detection of it more difficult. A significant number of newly discovered pieces of malware are published each day, and several automated toolkits for the creation of malware are readily accessible. Apart from the kind of analysis and machine learning techniques, the type of malware characteristics is the second key item in malware detection that matters much. The study provide a summary of the methods and tools available for detecting and analysing malware.

Malware is analysed statically and dynamically. Signature-based (antivirus) and behaviour-based anti-malware technologies are developed. Two difficulties plague signature-based approaches. New viruses are often undetectable by signature-based methods. Different strains of malware are able to evade security measures. Dynamic strategies are more resistant to malware concealment, while behavior-based approaches can identify new and variant infections. Signature-based algorithms identify known malware fast and effectively, whereas dynamic approaches are inflexible and time-consuming to develop. Overall, the accuracy and efficiency of malware detection have been greatly enhanced by machine and deep learning technologies. These strategies have also allowed for the creation of novel methods of identifying APTs and other forms of malware.

As per the findings, a more effective malware detection system may be developed employing both static and dynamic methodologies using machine learning approaches. Using machine learning and deep learning, this study conducts a comprehensive review on malware characteristics and classification methods as depicted in figure 6 and figure 7. Methodology employing N-gram, API, op-code and RT features utilizing various methods and machine learning algorithms provides better results as shown in figure 7, achieve maximum accuracy of 99.1% using Random Forest. This study can provide a comprehensive review to the researchers in the malware analysis field. In conclusion, this article has shown how the combination of machine and deep learning approaches has significantly advanced the field of malware detection. Machine learning and deep neural networks have become vital tools for keeping up with the ever-changing nature of cyber threats.

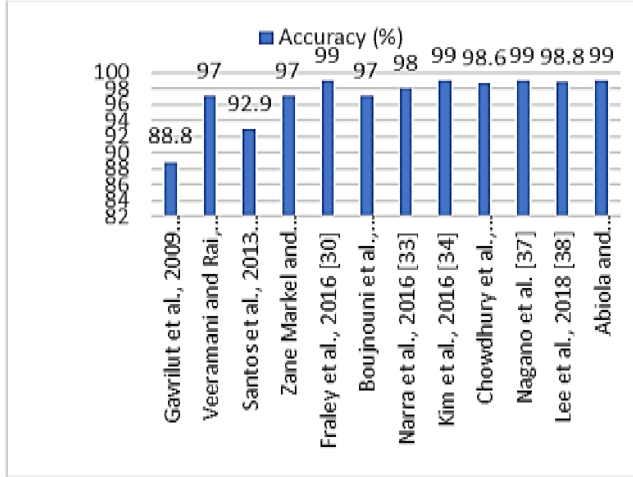


Fig. 4. Accuracy factors of signature-based detection

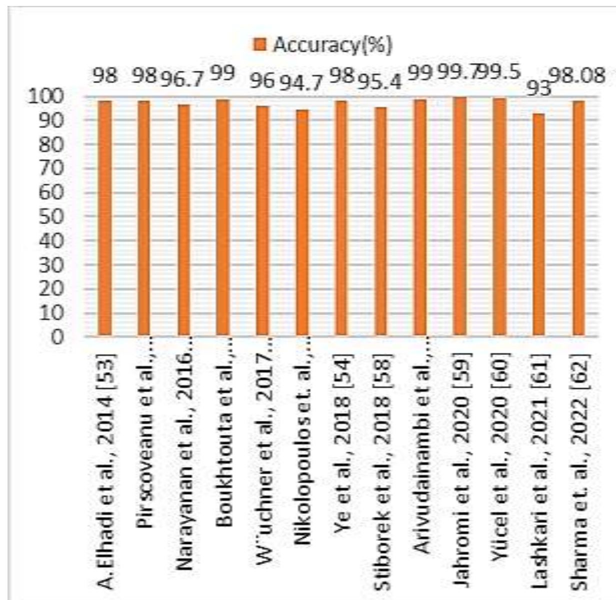
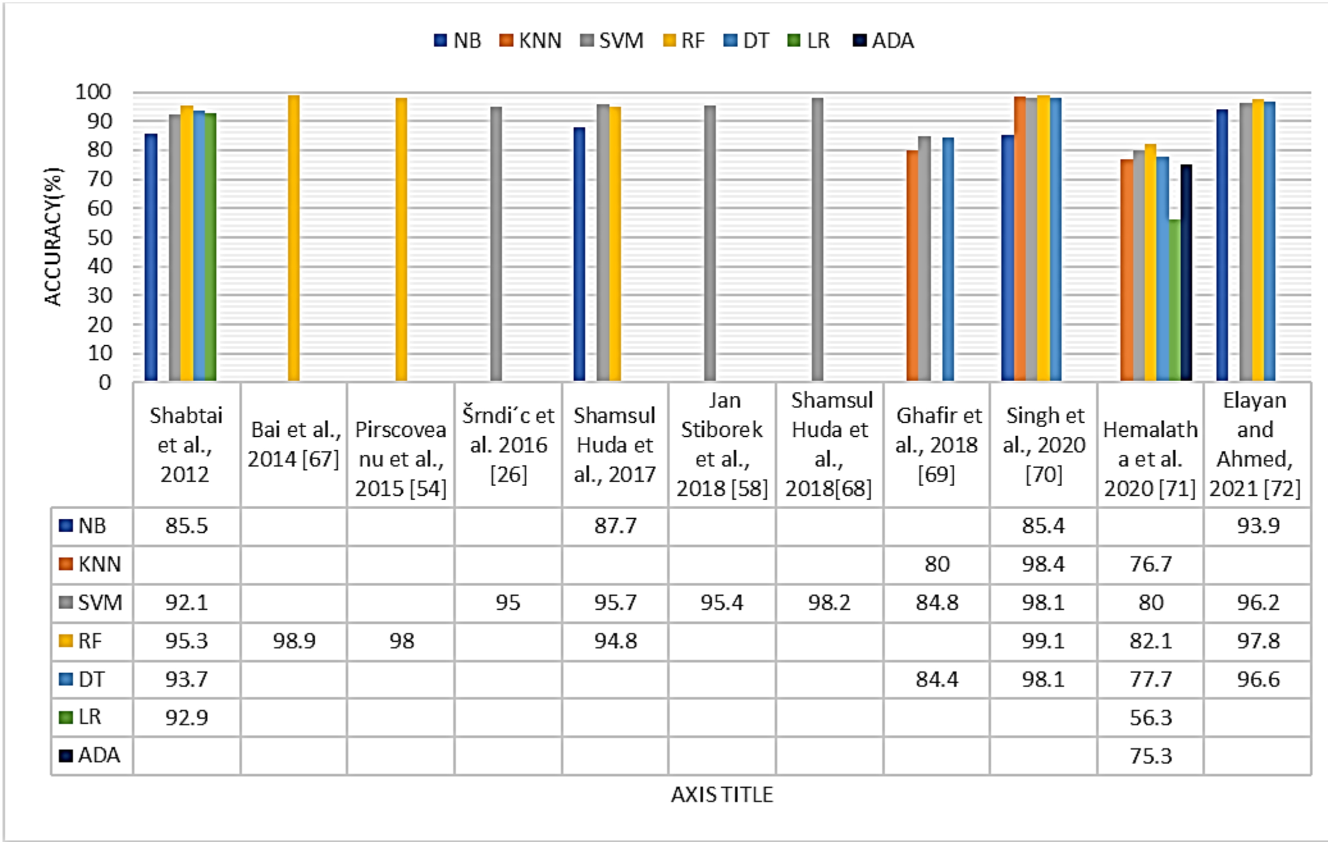


Fig. 5. Accuracy factors of behavior-based detection



	Shabtai et al., 2012	Bai et al., 2014 [67]	Pirscoveanu et al., 2015 [54]	Šrndi´c et al. 2016 [26]	Shamsul Huda et al., 2017	Jan Stiborek et al., 2018 [58]	Shamsul Huda et al., 2018[68]	Ghafir et al., 2018 [69]	Singh et al., 2020 [70]	Hemalatha et al. 2020 [71]	Elayan and Ahmed, 2021 [72]
NB	85.5				87.7				85.4		93.9
KNN								80	98.4	76.7	
SVM	92.1			95	95.7	95.4	98.2	84.8	98.1	80	96.2
RF	95.3	98.9	98		94.8				99.1	82.1	97.8
DT	93.7							84.4	98.1	77.7	96.6
LR	92.9									56.3	
ADA										75.3	

Fig. 6. Accuracy factors with respect to various ML algorithm

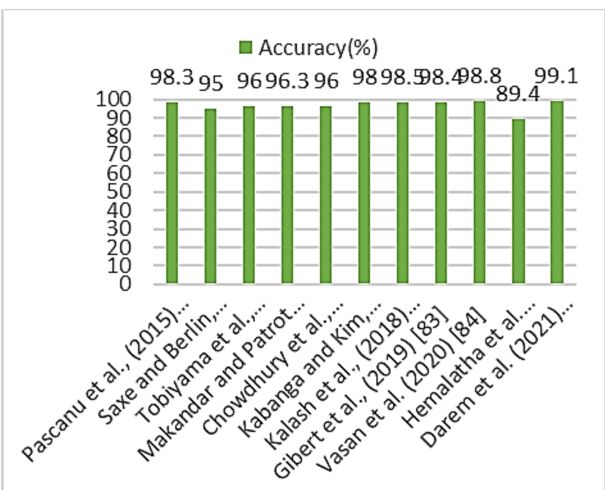


Fig. 7. Accuracy factors of Deep Learning approaches

Acknowledgment

The authors are grateful for the anonymous reviewers' and editor's insightful comments on the work, which allowed us to enhance both its quality and presentation.

References

[1] A. P. Namanya, A. Cullen, I. U. Awan, and J. P. Disso, "The World of Malware: An Overview," Proceedings - 2018 IEEE 6th International Conference on Future Internet of Things and Cloud, FiCloud 2018, pp. 420–427, 2018, doi: 10.1109/FiCloud.2018.00067.

[2] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, and L. Mao, "MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics," Computers and Security, vol. 83, pp. 208–233, 2019, doi: 10.1016/j.cose.2019.02.007.

[3] P. Burnap, R. French, F. Turner, and K. Jones, "Malware classification using self organising feature maps and machine activity data," Computers and Security, vol. 73, pp. 399–410, 2018, doi: 10.1016/j.cose.2017.11.016.

[4] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," Journal of Computer Security, vol. 19, no. 4, pp. 639–668, 2011, doi: 10.3233/JCS-2010-0410.

[5] P. Dwivedi and H. Sharan, "Analysis and Detection of Evolutionary Malware: A Review," International Journal of Computer Applications, vol. 174, no. 20, pp. 42–45, 2021, doi: 10.5120/ijca2021921005.

[6] X. Hu, "Large-Scale Malware Analysis, Detection, and Signature Generation," ProQuest Dissertations and Theses, p. 190, 2011, [Online]. Available: http://search.proquest.com/docview/918832186?accountid=44888

[7] S. Alam, R. N. Horspool, and I. Traore, "MARD: A framework for metamorphic malware analysis and real-time detection," Proceedings - International Conference on Advanced Information Networking and Applications, AINA, pp. 480–489, 2014, doi: 10.1109/AINA.2014.59.

[8] 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, doi: 10.1007/s11416-015-0261-z.

[9] E. Gandotra, D. Bansal, and S. Sofat, "Malware Analysis and Classification: A Survey," Journal of Information Security, vol. 05, no. 02, pp. 56–64, 2014, doi: 10.4236/jis.2014.52006.

[10] R. Kaur and M. Singh, "Hybrid Real-time Zero-day Malware Analysis and Reporting System," International Journal of Information Technology and Computer Science, vol. 8, no. 4, pp. 63–73, 2016, doi: 10.5815/ijitcs.2016.04.08.

[11] H. Gunduz, Y. Yaslan, and Z. Cataltepe, "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations," Knowledge-Based Systems, vol. 137, pp. 138–148, 2017, doi: 10.1016/j.knosys.2017.09.023.

[12] R. Veeramani and N. Rai, "Windows API based Malware Detection and Framework Analysis," … Conference on Networks and Cyber Security, vol. 3, no. 3, pp. 1–6, 2012, [Online]. Available: http://www.academia.edu/download/30183099/Conference_Proceedings_book_with_links.pdf#page=45

[13] M. Christodorescu, S. Jha, J. Kinder, S. Katzenbeisser, and H. Veith, "Software transformations to improve malware detection," Journal in Computer Virology, vol. 3, no. 4, pp. 253–265, 2007, doi: 10.1007/s11416-007-0059-8.

[14] N. Kawaguchi and K. Omote, "Malware function classification using apis in initial behavior," Proceedings - 2015 10th Asia Joint Conference on Information Security, AsiaJCIS 2015, pp. 138–144, 2015, doi: 10.1109/AsiaJCIS.2015.15.

[15] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," Human-centric Computing and Information Sciences, vol. 8, no. 1. 2018. doi: 10.1186/s13673-018-0125-x.

[16] M. Ijaz, M. H. Durad, and M. Ismail, "Static and Dynamic Malware Analysis Using Machine Learning," Proceedings of 2019 16th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2019, pp. 687–691, 2019, doi: 10.1109/IBCAST.2019.8667136.

[17] E. Raff et al., "An investigation of byte n-gram features for malware classification," Journal of Computer Virology and Hacking Techniques, vol. 14, no. 1, pp. 1–20, 2018, doi: 10.1007/s11416-016-0283-1.

[18] Y. Ding, X. Xia, S. Chen, and Y. Li, "A malware detection method based on family behavior graph," Computers and Security, vol. 73, pp. 73–86, 2018, doi: 10.1016/j.cose.2017.10.007.

[19] S. Kilgallon, L. de La Rosa, and J. Cavazos, "Improving the effectiveness and efficiency of dynamic malware analysis with machine learning," Proceedings - 2017 Resilience Week, RWS 2017, pp. 30–36, 2017, doi: 10.1109/RWEEK.2017.8088644.

[20] M. Al-Asli and T. A. Ghaleb, "Review of signature-based techniques in antivirus products," 2019 International Conference on Computer and Information Sciences, ICCIS 2019, pp. 1–6, 2019, doi: 10.1109/ICCISci.2019.8716381.

[21] S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A Survey on malware analysis and mitigation techniques," Computer Science Review, vol. 32, pp. 1–23, 2019, doi: 10.1016/j.cosrev.2019.01.002.

[22] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," Future Generation Computer Systems, vol. 90, pp. 211–221, 2019, doi: 10.1016/j.future.2018.07.052.

[23] J. Sexton, C. Storlie, and B. Anderson, "Subroutine based detection of APT malware," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 4, pp. 225–233, 2016, doi: 10.1007/s11416-015-0258-7.

[24] G. Laurenza, L. Aniello, R. Lazzeretti, and R. Baldoni, "Malware triage based on static features and public APT reports," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10332 LNCS, pp. 288–305, 2017, doi: 10.1007/978-3-319-60080-2_21.

[25] Y.-H. Choi, M.-Y. Jung, and S.-W. Seo, "L+1-MWM: A Fast Pattern Matching Algorithm for High-Speed Packet Filtering," pp. 2288–2296, 2008, doi: 10.1109/infocom.2008.297.

[26] N. Šrndić and P. Laskov, "Hidost: a static machine-learning-based detector of malicious files," Eurasip Journal on Information Security, vol. 2016, no. 1, pp. 1–20, 2016, doi: 10.1186/s13635-016-0045-0.

[27] B. Wu, X. Lin, W. D. Li, T. L. Lu, and D. M. Zhang, "Smartphone malware detection model based on artificial immune system in cloud computing," Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications, vol. 38, no. 4, pp. 33–37, 2015, doi: 10.13190/j.jbupt.2015.04.008.

[28] J. Wang, G. Li, and J. Fe, "Fast-join: An efficient method for fuzzy token matching based string similarity join," Proceedings - International Conference on Data Engineering, pp. 458–469, 2011, doi: 10.1109/ICDE.2011.5767865.

[29] A. Mohaisen, O. Alrawi, and M. Mohaisen, "AMAL: High-fidelity, behavior-based automated malware analysis and classification," Computers and Security, vol. 52, pp. 251–266, 2015, doi: 10.1016/j.cose.2015.04.001.

[30] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pp. 38–49, 2001, doi: 10.1109/secpri.2001.924286.

[31] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of Internet malware," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4637 LNCS, pp. 178–197, 2007, doi: 10.1007/978-3-540-74320-0_10.

[32] M. Eskandari, Z. Khorshidpour, and S. Hashemi, "HDM-Analyser: A hybrid analysis approach based on data mining techniques for malware detection," Journal in Computer Virology, vol. 9, no. 2, pp. 77–93, 2013, doi: 10.1007/s11416-013-0181-8.

[33] S. Sheen, R. Anitha, and V. Natarajan, "Android based malware detection using a multifeature collaborative decision fusion approach," Neurocomputing, vol. 151, no. P2, pp. 905–912, 2015, doi: 10.1016/j.neucom.2014.10.004.

[34] B. N. Narayanan, O. Djaneye-Boundjou, and T. M. Kebede, "Performance analysis of machine learning and pattern recognition algorithms for Malware classification," Proceedings of the IEEE National Aerospace Electronics Conference, NAECON, vol. 0, pp. 338–342, 2016, doi: 10.1109/NAECON.2016.7856826.

[35] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 2, pp. 59–67, 2016, doi: 10.1007/s11416-015-0244-0.

[36] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning," Tsinghua Science and Technology, vol. 21, no. 1, pp. 114–123, 2016, doi: 10.1109/TST.2016.7399288.

[37] J. Ming, Z. Xin, P. Lan, D. Wu, P. Liu, and B. Mao, "Impeding behavior-based malware analysis via replacement attacks to malware specifications," Journal of Computer Virology and Hacking Techniques, vol. 13, no. 3, pp. 193–207, 2017, doi: 10.1007/s11416-016-0281-3.

[38] S. D. Nikolopoulos and I. Polenakis, "A graph-based model for malware detection and classification using system-call groups," Journal of Computer Virology and Hacking Techniques, vol. 13, no. 1, pp. 29–46, 2017, doi: 10.1007/s11416-016-0267-1.

[39] A. Altaher, "An improved Android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features," Neural Computing and Applications, vol. 28, no. 12, pp. 4147–4157, 2017, doi: 10.1007/s00521-016-2708-7.

[40] D. Arivudainambi, V. K. Varun, S. C. S., and P. Visu, "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance," Computer Communications, vol. 147, no. July, pp. 50–57, 2019, doi: 10.1016/j.comcom.2019.08.003.

[41] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, "A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing," Journal of Network and Computer Applications, vol. 151, p. 102507, 2020, doi: 10.1016/j.jnca.2019.102507.

[42] Baset Mohamad, "Machine Learning for Malware Detection," Msc Projects, no. November, pp. 1–17, 2016, [Online]. Available: www.kaspersky.com

[43] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," Journal of Network and Computer Applications, vol. 153, p. 102526, 2020, doi: 10.1016/j.jnca.2019.102526.

[44] R. Sinha, "Study of malware detection using machine learning," no. July, 2021, doi: 10.13140/RG.2.2.11478.16963.

[45] A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, "Detecting unknown malicious code by applying classification techniques on OpCode patterns," Security Informatics, vol. 1, no. 1, pp. 1–22, 2012, doi: 10.1186/2190-8532-1-1.

[46] R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, and J. M. Pedersen, "Analysis of malware behaviour classification from ML," 2015.

[47] J. Bai, J. Wang, and G. Zou, "A malware detection scheme based on mining format information," Scientific World Journal, vol. 2014, 2014, doi: 10.1155/2014/260905.

[48] J. Singh and J. Singh, "Assessment of supervised machine learning algorithms using dynamic API calls for malware detection," International Journal of Computers and Applications, vol. 44, no. 3, pp. 270–277, 2022, doi: 10.1080/1206212X.2020.1732641.

[49] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet‐based deep learning model for Malware detection," Entropy, vol. 23, no. 3, pp. 1–23, 2021, doi: 10.3390/e23030344.

[50] J. Stiborek, T. Pevný, and M. Rehák, "Multiple instance learning for malware classification," Expert Systems with Applications, vol. 93, pp. 346–357, 2018, doi: 10.1016/j.eswa.2017.10.036.

[51] M. Goyal and R. Kumar, "A Survey on Malware Classification Using Machine Learning and Deep Learning," International Journal of Computer Networks and Applications, vol. 8, no. 6, pp. 758–775, 2021, doi: 10.22247/ijcna/2021/210724.

[52] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," IEEE Access, vol.

7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[53] Q. Le, O. Boydell, B. mac Namee, and M. Scanlon, "Deep learning at the shallow end: Malware classification for non-domain experts," Proceedings of the Digital Forensic Research Conference, DFRWS 2018 USA, pp. S118–S126, 2018, doi: 10.1016/j.diin.2018.04.024.

[54] 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, doi: 10.1007/s10115-017-1058-9.

[55] H. Rathore, S. Agarwal, S. K. Sahay, and M. Sewak, Malware detection using machine learning and deep learning, vol. 11297 LNCS. Springer International Publishing, 2018. doi: 10.1007/978-3-030-04780-1_28.

[56] M. Ashik et al., "Detection of malicious software by analyzing distinct artifacts using machine learning and deep learning algorithms," Electronics (Switzerland), vol. 10, no. 14, pp. 1–28, 2021, doi: 10.3390/electronics10141694.

[57] A. Namavar Jahromi et al., "An improved two-hidden-layer extreme learning machine for malware hunting," Computers and Security, vol. 89, p. 101655, 2020, doi: 10.1016/j.cose.2019.101655.

[58] S. Huda, R. Islam, J. Abawajy, J. Yearwood, M. M. Hassan, and G. Fortino, "A hybrid-multi filter-wrapper framework to identify run-time behaviour for fast malware detection," Future Generation Computer Systems, vol. 83, pp. 193–207, 2018, doi: 10.1016/j.future.2017.12.037.

[59] S. Huda et al., "Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data," Information Sciences, vol. 379, pp. 211–228, 2017, doi: 10.1016/j.ins.2016.09.041.

[60] A. A. E. Elhadi, M. A. Maarof, B. I. A. Barry, and H. Hamza, "Enhancing the detection of metamorphic malware using call graphs," Computers and Security, vol. 46, pp. 62–78, 2014, doi: 10.1016/j.cose.2014.07.004.

[61] A. Boukhtouta, S. A. Mokhov, N. E. Lakhdari, M. Debbabi, and J. Paquet, "Network malware classification comparison using DPI and flow packet headers," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 2, pp. 69–100, 2016, doi: 10.1007/s11416-015-0247-x.

[62] T. Wuchner, A. Cislak, M. Ochoa, and A. Pretschner, "Leveraging compression-based graph mining for behavior-based malware detection," IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 1, pp. 99–112, 2019, doi: 10.1109/TDSC.2017.2675881.

[63] Ç. Yücel and A. Koltuksuz, "Imaging and evaluating the memory access for malware," Forensic Science International: Digital Investigation, vol. 32, 2020, doi: 10.1016/j.fsidi.2019.200903.

[64] A. H. Lashkari, B. Li, T. L. Carrier, and G. Kaur, "VolMemLyzer: Volatile Memory Analyzer for Malware Classification using Feature Engineering," 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge, RDAAPS 2021, no. Cic, 2021, doi: 10.1109/RDAAPS48126.2021.9452028.

[65] S. Sharma, C. R. Krishna, and R. Kumar, "RansomDroid: Forensic analysis and detection of Android Ransomware using unsupervised machine learning technique," Forensic Science International: Digital Investigation, vol. 37, p. 301168, 2021, doi: 10.1016/j.fsidi.2021.301168.

[66] D. Gavriluţ, M. Cimpoeşu, D. Anton, and L. Ciortuz, "Malware detection using machine learning," Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT '09, vol. 4, pp. 735–741, 2009, doi: 10.1109/IMCSIT.2009.5352759.

[67] I. Santos, F. Brezo, X. Ugarte-Pedrero, and P. G. Bringas, "Opcode sequences as representation of executables for data-mining-based unknown malware detection," Information Sciences, vol. 231, pp. 64–82, 2013, doi: 10.1016/j.ins.2011.08.020.

[68] Z. Markel and M. Bilzor, "Building a machine learning classifier for malware detection," WATeR 2014 - Proceedings of the 2014 2nd Workshop on Anti-Malware Testing Research, 2015, doi: 10.1109/WATeR.2014.7015757.

[69] J. B. Fraley, "Polymorphic Malware Detection Using Topological Feature Extraction with Data Mining," 2016.

[70] M. el Boujnouni, M. Jedra, and N. Zahid, "New malware detection framework based on N-grams and Support Vector Domain Description," Proceedings of the 2015 11th International Conference on Information Assurance and Security, IAS 2015, pp. 123–128, 2016, doi: 10.1109/ISIAS.2015.7492756.

[71] U. Narra, F. di Troia, V. A. Corrado, T. H. Austin, and M. Stamp, "Clustering versus SVM for malware detection," Journal of Computer Virology and Hacking Techniques, vol. 12, no. 4, pp. 213–224, 2016, doi: 10.1007/s11416-015-0253-z.

[72] Dong-Hee Kim, S.-U. Woo, D.-K. Lee, and and T.-M. Chung, "Static detection of malware and benign executable using machine learning algorithm," no. c, pp. 14–19, 2016.

[73] M. Chowdhury, A. Rahman, and R. Islam, "Malware analysis and detection using data mining and machine learning classification," Advances in Intelligent Systems and Computing, vol. 580, no. January, pp. 266–274, 2018, doi: 10.1007/978-3-319-67071-3_33.

[74] Y. Nagano and R. Uda, "Static analysis with paragraph vector for malware detection," Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, IMCOM 2017, 2017, doi: 10.1145/3022227.3022306.

[75] A. M. Abiola and M. F. Marhusin, "Signature-based malware detection using sequences of N-grams," International Journal of Engineering and Technology(UAE), vol. 7, no. 4, pp. 120–125, 2018, doi: 10.14419/ijet.v7i4.15.21432.

[76] I. Ghafir et al., "Detection of advanced persistent threat using machine-learning correlation analysis," Future

Generation Computer Systems, vol. 89, pp. 349–359, 2018, doi: 10.1016/j.future.2018.06.055.

[77] O. N. Elayan and A. M. Mustafa, "Android malware detection using deep learning," Procedia Computer Science, vol. 184, no. 2019, pp. 847–852, 2021, doi: 10.1016/j.procs.2021.03.106.

[78] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 2015-Augus, pp. 1916–1920, 2015, doi: 10.1109/ICASSP.2015.7178304.

[79] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," 2015 10th International Conference on Malicious and Unwanted Software, MALWARE 2015, pp. 11–20, 2016, doi: 10.1109/MALWARE.2015.7413680.

[80] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware Detection with Deep Neural Network Using Process Behavior," Proceedings - International Computer Software and Applications Conference, vol. 2, pp. 577–582, 2016, doi: 10.1109/COMPSAC.2016.151.

[81] A. Makandar and A. Patrot, "Malware analysis and classification using Artificial Neural Network," International Conference on Trends in Automation, Communication and Computing Technologies, I-TACT 2015, 2016, doi: 10.1109/ITACT.2015.7492653.

[82] E. K. Kabanga and C. H. Kim, "Malware Images Classification Using Convolutional Neural Network," Journal of Computer and Communications, vol. 06, no. 01, pp. 153–158, 2018, doi: 10.4236/jcc.2018.61016.

[83] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, "Malware Classification with Deep Convolutional Neural Networks," 2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018 - Proceedings, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/NTMS.2018.8328749.

[84] 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, doi: 10.1007/s11416-018-0323-0.

[85] 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, doi: 10.1016/j.comnet.2020.107138.

[86] A. Darem, J. Abawajy, A. Makkar, A. Alhashmi, and S. Alanazi, "Visualization and deep-learning-based malware variant detection using OpCode-level features," Future Generation Computer Systems, vol. 125, pp. 314–323, 2021, doi: 10.1016/j.future.2021.06.032