International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Intrusion Detection System Using Various Machine Learning Approaches with Ensemble Learning: A Systematic Review

Ms. Pragati V. Pandit, Research Scholar, SJJTU Jhunjhunu, Rajasthan, pragativpandit2918@gmail.com

Dr. Shashi Bhushan, Research Guid, SJJTU Jhunjhunu, Rajasthan, tyagi_shashi@yahoo.com

Dr. Uday Patkar, Research Co-Guid, SJJTU Jhunjhunu, Rajasthan, udaypatkarcomp"gmail.com

ARTICLEINFO

Article History:

Accepted: 10 June 2023
Published: 29 June 2023

Publication Issue
Volume 10, Issue 3
May-June-2023

Page Number
1207-1216

ABSTRACT

Although feature filtering-based network intrusion detection systems have some drawbacks that make it difficult for security managers and analysts to detect and stop network intrusions in their enterprises, recent years have seen a surge in advanced threat attacks. Using methods for detecting intrusions, information systems are routinely safeguarded and damage is reduced. It safeguards against threats and flaws in physical and virtual computer networks. Machine learning techniques are now being used to build efficient intrusion detection systems. Machine learning techniques for intrusion detection include rule learning, ensemble techniques, statistical models, and neural networks. Techniques used in machine learning ensembles are recognised for performing exceptionally well during the learning process. A suitable ensemble technique needs to be researched in order to build a successful intrusion detection system. In this research, we combined the decision tree, random forest, extra tree, and XGBoost algorithms with a novel ensemble method for intrusion detection in the network. The suggested approach enhances detection precision and was developed using the Python computer language. The developed system is assessed using the CICIDS2017 dataset according to a number of evaluation criteria, such as precision, recall, and f1-score. The detection accuracy is greatly improved by the ensemble approach.

Keywords—Intrusion detection, Machine Learning, ML algorithms, ensemble learning, Random Forest, Decision Tree, XgBoost, Extra Tree.

I. INTRODUCTION

Viruses, worms, ransomware, trojan horses, and spyware are all considered to be malware or malicious software. Malware is an intentionally introduced set of codes that interfere with the operation of computing devices in modern advanced applications. The two types of malware detection methods are signature-based and behaviour based methods. These malware analysis methods include static, dynamic, and hybrid approaches. Machine learning is a novel method of malware detection that needs trained datasets to identify malware. Therefore, it is suggested in the current situation to combine data mining and machine learning approaches to identify malware that uses obfuscation and polymorphism tactics to conceal. Due to issues with the effectiveness of learning models, the notion of malware detection has been extensively investigated in the modern era. Several methods are used to critically assess the deep learning proposals made using various data mining approaches. In the Internet- of-Things era, malware poses a serious threat to smart computer devices (IoT). With the rising use of IoT-based smart devices including computers, mobile phones, data servers, tablets, and other equipment, it has spread rapidly.

Drive-by-download malware can cause harm to the target system [2] and present issues with data security and data privacy. Through the internet, hackers can affect crucial infrastructures [3], [4], [5], and IoT devices [6]. Due to the rising usage of the internet on smart devices nowadays, malware has been exposed in a wide range of system attacks. Figure 1 depicts the current situation of cyberattacks and the corresponding avenue loss. Cost increases over time illustrate the expense of losses brought on by malware that is embedded in apps. Malware's capacity to be moved around allows it to spread alarmingly quickly across a variety of operating systems. Figure 1 below, from a recent study on cyber losses, details the effects of malware on company, information, and equipment damage, as well as income loss brought on by malware. With an increase in revenue loss throughout the years, it has incorporated 355 enterprises across 16 industrial sectors from 11 different nations. [7] As a result, malware identification plays a big part in internet security.

Researchers are taking note of malware detection since it has a wide range of practical domain solutions. It is difficult to use cutting-edge technologies to work with a crucial malware function. Numerous investigations are carried out to evaluate the approaches suggested to find malware with various classifications. Fig. 2 displays a timeline of articles that were published throughout the previous few years. It has been noted that study on malware is constantly growing as people become more aware of the importance of the topic.

With the widespread use of the internet and mobile devices, the demand for the topic has increased. It suggests that in recent years, problems brought on by malware detection have gotten worse. Despite an increase in research on malware detection, cutting-edge technologies have not yet attempted to develop reliable malware recognition methods.

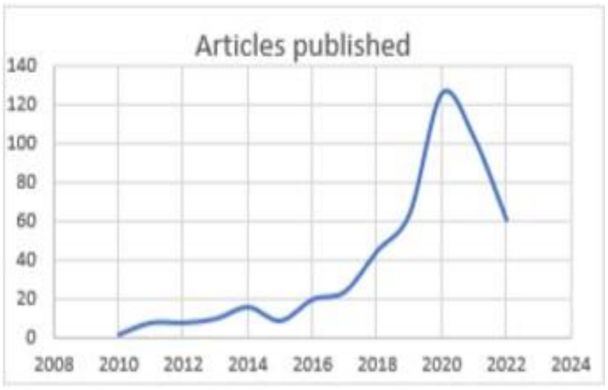


Fig. 2: Articles published in recent years

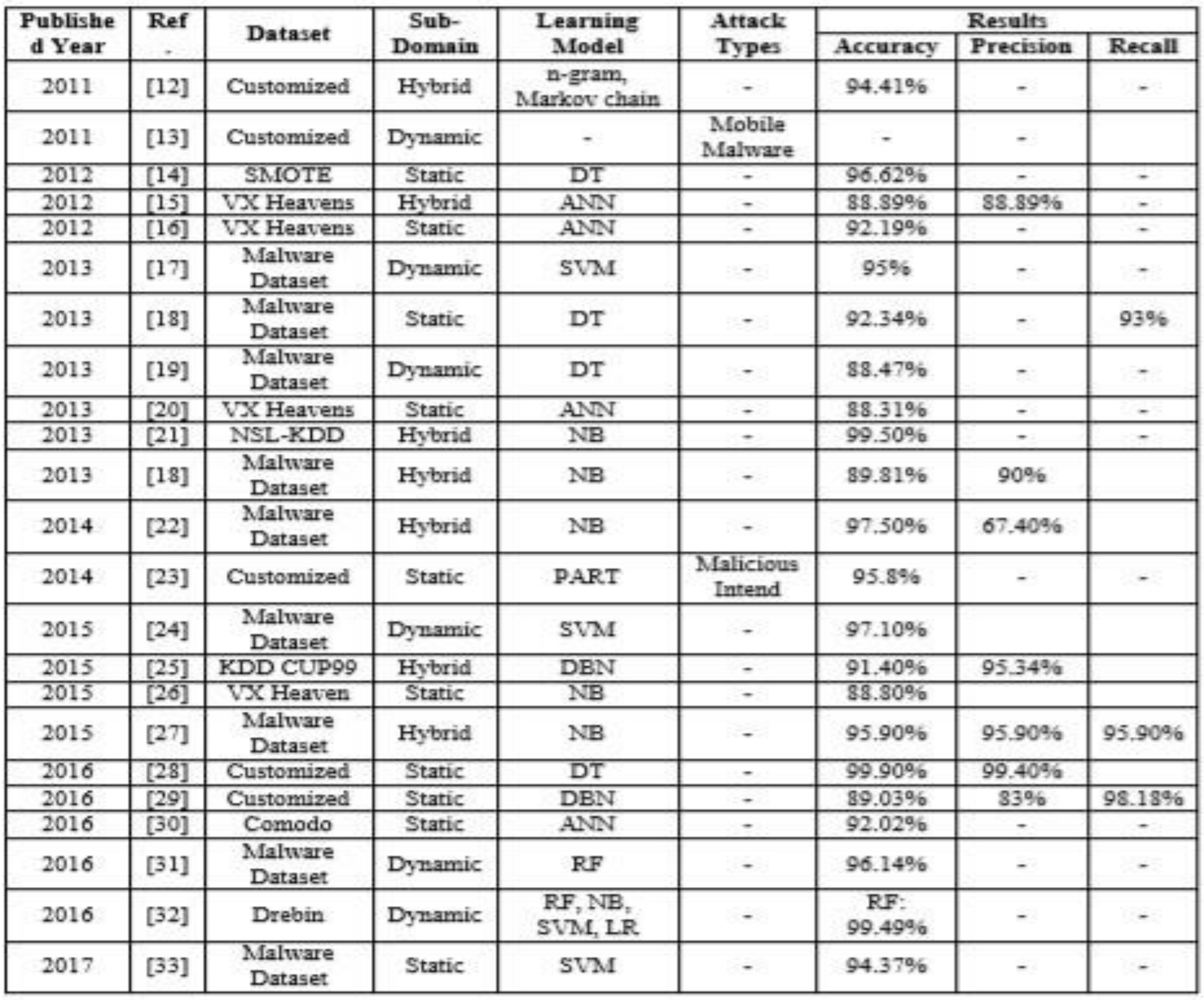
II. LITERATURE REVIEW

A review of various methods for malware detection revealed that signature-based strategies were primarily employed in the past. Additionally, it has been noted that the effectiveness of these techniques in identifying novel or zero-day malware threats is subpar. However, machine learning approaches surpass other methods in detecting zero-day malware attacks and are adept at spotting them. [9]

We can compare techniques in terms of usages and performance because the various strategies employed in ML classification have varying degrees of performance. A decision tree is used for malware detection 29% less frequently than SVM approaches, which are employed 29% more frequently. DBN is combined with semi-supervised learning approaches to increase detection technique accuracy.

The detailed explanation in the table below illustrates the methods utilized for malware detection throughout the previous ten years. [11] Table 1 is compared with other datasets that suggest greater accuracy in Table 2 below. While Jamil Q et altailored dataset was detected with a maximum accuracy of 99.90%, Z. Ma et alexplanation .'s of the Android Malware dataset had a greater accuracy of 97.22%. Similar to how accuracy of datasets like Contagio, Contagio Dump, VirusShare, Drebin, Moledroid Apps, NSL-KDD, etc. ranges around 99%, accuracy of the KDD CUP99 dataset, however, is less accurate and only reaches 91.40%, according to Li Y et al. In Fig. 3, it is further illustrated graphically. The performance and accuracy of our suggested model are to be enhanced.

Table 1 : Comparison of studies in different datasets



Publishe d Year	Ref .	Dataset	Sub-Domain	Learning Model	Attack Types	Results		
						Accuracy	Precision	Recall
2011	[12]	Customized	Hybrid	n-gram, Markov chain	-	94.41%	-	-
2011	[13]	Customized	Dynamic	-	Mobile Malware	-	-	
2012	[14]	SMOTE	Static	DT	-	96.62%	-	-
2012	[15]	VX Heavens	Hybrid	ANN	-	88.89%	88.89%	-
2012	[16]	VX Heavens	Static	ANN	-	92.19%	-	-
2013	[17]	Malware Dataset	Dynamic	SVM	-	95%	-	-
2013	[18]	Malware Dataset	Static	DT	-	92.34%	-	93%
2013	[19]	Malware Dataset	Dynamic	DT	-	88.47%	-	-
2013	[20]	VX Heavens	Static	ANN	-	88.31%	-	-
2013	[21]	NSL-KDD	Hybrid	NB	-	99.50%	-	-
2013	[18]	Malware Dataset	Hybrid	NB	-	89.81%	90%	
2014	[22]	Malware Dataset	Hybrid	NB	-	97.50%	67.40%	
2014	[23]	Customized	Static	PART	Malicious Intend	95.8%	-	-
2015	[24]	Malware Dataset	Dynamic	SVM	-	97.10%		
2015	[25]	KDD CUP99	Hybrid	DBN	-	91.40%	95.34%	
2015	[26]	VX Heaven	Static	NB	-	88.80%		
2015	[27]	Malware Dataset	Hybrid	NB	-	95.90%	95.90%	95.90%
2016	[28]	Customized	Static	DT	-	99.90%	99.40%	
2016	[29]	Customized	Static	DBN	-	89.03%	83%	98.18%
2016	[30]	Comodo	Static	ANN	-	92.02%	-	-
2016	[31]	Malware Dataset	Dynamic	RF	-	96.14%	-	-
2016	[32]	Drebin	Dynamic	RF, NB, SVM, LR	-	RF: 99.49%	-	-
2017	[33]	Malware Dataset	Static	SVM	-	94.37%	-	-

Table 2: Comparison of datasets and accuracy

Hou S et al. [38]	Comodo	96.66%
Nix R et al. [37]	Contagio	99.40%
A. Mehtab et al. [50]	Contagio Dump, VirusShare	99.11%
Jamil Q et al. [28]	Customized	99.90%
X. Pei et al. [54]	Drebin	99.69%
Li Y et al. [25]	KDD CUP99	91.40%
H. Naeem et al. [53]	Leopard Mobile dataset	98.79%
Salehi Z et al. [22]	Malware Dataset	97.50%
Mosli R et al. [55]	Moledroid Apps	99.10%
Amjad Hussain et al. [21]	NSL-KDD	99.50%
Kapravelos A et al. [14]	SMOTE	96.62%
Karbab EMB et al. [44]	VirusShare	98.29%
Shabtai A et al. [16]	VX Heavens	92.19%

Various models are represented in table 3 along with an explanation of how they work. To evaluate each model's performance and choose the best one, these models are compared. Each model has drawbacks, which are examined in order to get around them utilising this approach. Additionally, other studies have been conducted on these models in recent years to support varioustechniques for improved results.

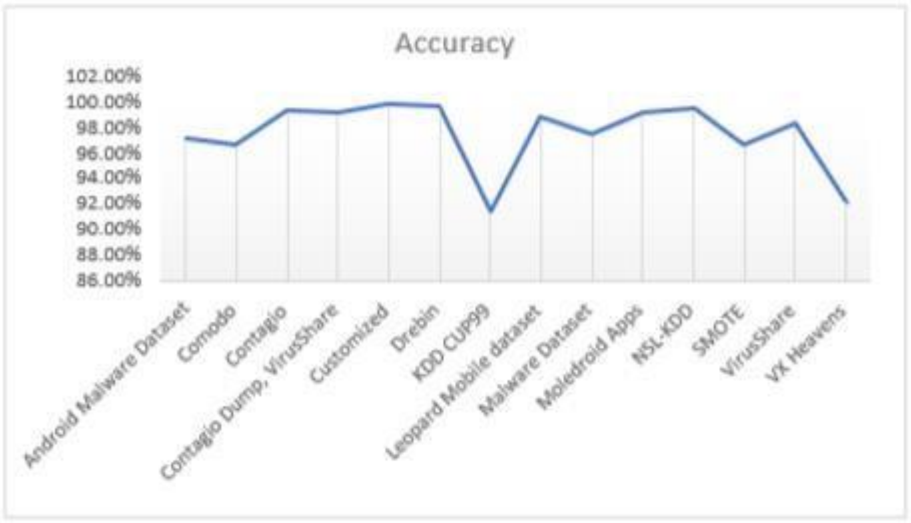


Fig. 3: Accuracy of different datasets obtained in different models Table 3: Functioning of Models and Their Limitations

Model	Year	Ref.	Description	Limitations
SVM	1995	[56]	• Used for regression and classification • Reduced overfitting issue	• Impotent to efficient handling of big or noisier datasets. • Costly computational process.
Rando m Forest	1995	[57]	• Integrated with many DTs. • Each DT produces a prediction. • Final prediction has a maximum number of votes in the model.	• Costly computational process. • Slower prediction generation.
Naïve Bayes	1960	[58]	• A probabilistic classifier with rapid computational process. • A feature is adopted as entirely independent of all other current features.	• Assigns 0 probabilities for some absent test data set category in the training data set. • Stores all training samples • Requires enormous data for good results.
RNN	1982	[59]	• Efficiently models sequential data • Quickly memorize the sequential events • Different various, i.e. LSTM is available	• Difficult training of the network.

III. DISCUSSION

Precision = TP/ (TP+FP) (eq. 1)

Malware detection involves the knowledge of cryptic malware protection as a fundamental component of machine learning tactics due to the emerging malware in the innovation. [65] The two groups of machine learning techniques—supervised and unsupervised—allow for the employment of the appropriate technique or, in some Circumstances, a semi-supervised technique. Malicious applications, or malware, are detected using behavior-based and signature-based methods [66] that employ static and dynamic malware analysis [67], [68]. A malware detection taxonomy for aspects of the API calls, assembly, and binary features is illustrated by machine learning techniques. Additionally, these traits are important for machine learning techniques that predict and identify malware.

Precision

Ratio of correctly classified benign or positive samples or applications to all correctly classified benign or positive samples or applications in the dataset is how precision is calculated (see the eq. 1). A greater precision value can result in an excellent performance. Recall is frequently referred to as a "true positive rate," which is a proportion of correctly identified benign or positive samples or applications to all benign or positive samples or applications in the dataset (see the eq. 2). A classifier only performs well when the recall value is large Recall = TP/ (TP+FN) (eq. 2)

Accuracy

Accuracy is determined by the percentage of samples or applications in a dataset that are correctly categorised (see eq. 3). Since accuracy determines how accurate the classifier is, accuracy should have a greater number for better performance.

Accuracy= (TP+TN)/ (TN+FP+FN+TP) (eq.3)

Distinct accuracies have been attained in recent years, according to research done on various datasets. Similar datasets are used to examine various malware assaults in various subdomains, such as DT, ANN, SVM, etc. Comparing these research, it can be shown that datasets are continuously used with various models to produce results with improved precision and accuracy.

Since it helps identify both previously known and new attacks, the ensemble approach to intrusion detection has gained in significance. As a result, this recommended solution offers an ensemble-based IDS based on machine learning techniques. The block diagram of the proposed system depicts the numerous steps of the procedure in Figure

The CICIDS2017 dataset was used to conduct the experiment. These datasets are used to assess the effectiveness of the suggested approach. For the system we intended to create, Python was employed. The accuracy of the suggested system is evaluated by comparing the Decision Tree, Random Forest, Extra Tree, and XgBoost algorithms.

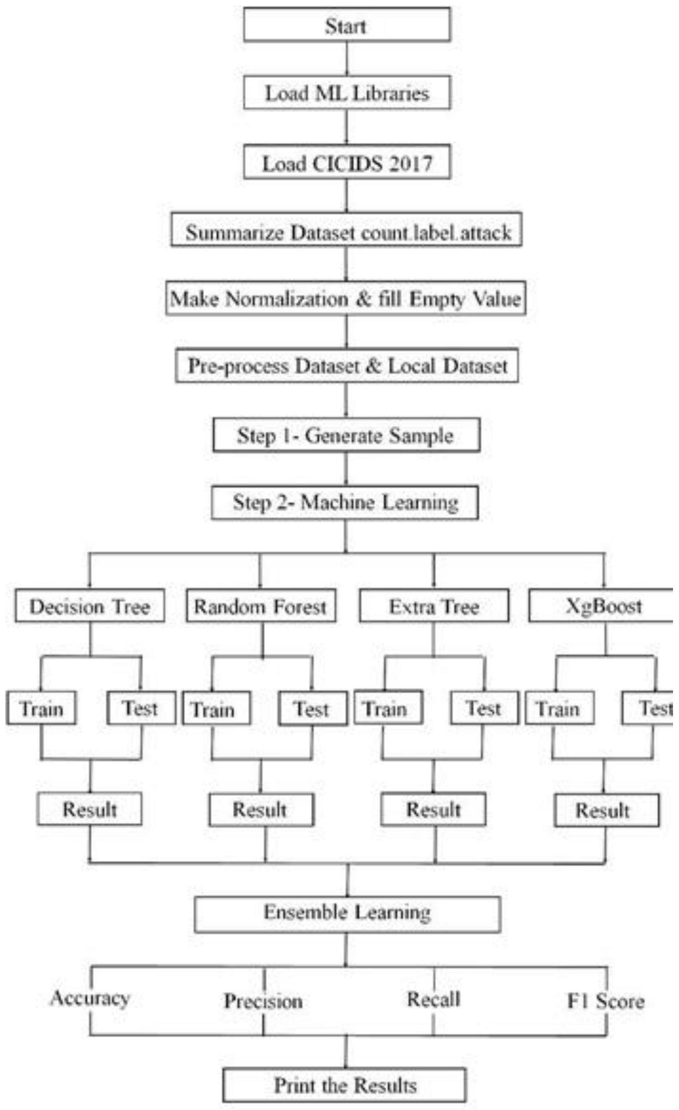


Fig 4: Overview of the Complete Framework

This method was previously used to build collaborative IDS, which allows attributes to be measured whatever they want and increases forecast accuracy. The suggested system's performance is evaluated using the listed parameters.

The accuracy parameter measures how well the classifier can distinguish between instances that are truly negative and those that are false negatives.

The classifier's ability to correctly identify a negative occurrence, not FN, is the parameter precision.

The value of recall is the classifier's capacity to correctly recognise each positive example.

The result obtained after combining the accuracy and recall values is known as the f1-score parameter.

The ML libraries are originally imported by our suggested intrusion detection system utilising Python programming. The following step involves reading the CICIDS2017 dataset. The dataset is then tested for intrusions. The empty or null values are filled with zero if the dataset has been detected to have been tampered with. The dataset is then trained and the pre-processing is completed. To identify system imbalances, one uses the SMOTE library. Then, using ensemble learning, the DT, RF, ET, and XgBoost algorithms test the dataset. The suggested system's algorithms separately train and test the dataset. The model collects data, chooses features, does pre-processing, and prepares the model for training, testing, and validation.

IV. CONCLUSION

In this study, we investigated various models built using various methodologies. Each model's accuracy is compared, and the suggested model is investigated using machine learning techniques. We monitored and classified malware samples using an RF classifier, then we measured the outcomes. To train the machine learning-based classifier, trained data is needed. The program's execution on system calls and function calls is noted. Due to its higher accuracy, this framework can identify cyber threats in networks and virtualized computer systems while overcoming false positives. In compared to other approaches, recall and precision are further countable metrics that can be measured. Based on static and dynamic analysis, the ensemble model characteristics for mobile applications and web browsers, the algorithm can identify various samples as benign or dangerous. Accurate real-world data sets can predict viruses and perform better in the security field.

V. REFERENCES

[1]. N. Sharma and B. Arora, "Data Mining and Machine Learning Techniques for Malware Detection," in Advances in Intelligent Systems and Computing, 2021, vol. 1187, pp. 557–567. doi: 10.1007/978-981-15-6014-9_66.

[2]. G. McGraw and G. Morrisett, "Attacking malicious code: A report to the Infosec Research Council," IEEE Softw., vol. 17, no. 5, pp. 33–41, 2000, doi: 10.1109/52.877857.

[3]. A. Kapravelos, Y. Shoshitaishvili, M. Cova, C. Kruegel, and G. Vigna, "Revolver: An automated approach to the detection of evasive web-based malware," in Proceedings of the 22nd USENIX Security Symposium, 2013, pp. 637–651. Accessed: Jul. 16, 2022.

[4]. R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," IEEE Secur. Priv., vol. 9, no. 3, pp. 49–51, May 2011, doi: 10.1109/MSP.2011.67.

[5]. E. Chien, L. OMurchu, N. F.-5th U. W. on Large-Scale, and U. 2012, "{W32. Duqu}: The Precursor to the Next Stuxnet," usenix.org, Accessed: Jul. 16, 2022.

[6]. E. Cozzi, M. Graziano, Y. Fratantonio, and D. Balzarotti, "Understanding Linux Malware," in Proceedings – IEEE. Symposium on Security and Privacy, 2018, vol. 2018-May, pp. 161–175. doi: 10.1109/SP.2018.00054.

[7]. K. Bissell and L. Ponemon, "The cost of cybercrime," Netw. Secur., vol. 2015, no. 10, p. 2, 2015, doi: 10.1016/s1353- 4858(15)30085-4.

[8]. Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," in Proceedings of the 2017 Design, Automation and Test in Europe, DATE 2017, 2017, pp. 169–174. doi: 10.23919/DATE.2017.7926977.

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

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

[11]. K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," IEEE Access, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[12]. J. Neil et al., "Graph-based malware detection using dynamic analysis," Springer, vol. 7, no. 4, pp. 247–258, Nov. 2011, doi: 10.1007/s11416-011-0152-x.

[13]. [13]T. Isohara, K. Takemori, and A. Kubota, "Kernel-based behavior analysis for android malware detection," in Proceedings - 2011 7th International Conference on Computational Intelligence and Security, CIS 2011, 2011, pp. 1011–1015. doi: 10.1109/CIS.2011.226.

[14]. [14] T. Kavzoglu and I. Colkesen, "The effects of training set size for performance of support vector machines and decision trees," gtu.edu.tr, pp. 127–132, 2009, Accessed: Jul. 18, 2022. [15] Y. Chen, A. Narayanan, S. Pang, and B. Tao, "Multiple sequence alignment and artificial neural networks for malicious software

detection," in Proceedings - International Conference on Natural Computation, 2012, pp. 261–265. doi: 10.1109/ICNC.2012.6234576.

[15]. A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, "Detecting unknown malicious code by applying classification techniques on OpCode patterns," Secur. Inform., vol. 1, no. 1, Dec. 2012, doi: 10.1186/2190-8532-1-1.

[16]. A. Mohaisen and O. Alrawi, "Unveiling zeus: automated classification of malware samples," May 2013, pp. 829–832. doi: 10.1145/2487788.2488056.

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

[18]. R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," Journal of Network and Computer Applications, vol. 36, no. 2. pp. 646–656, Mar. 2013. doi: 10.1016/j.jnca.2012.10.004.

[19]. C. Liangboonprakong and O. Sornil, "Classification of malware families based on Ngrams sequential pattern features," in Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013, 2013, pp. 777–782. doi:10.1109/ ICIEA.2013. 6566472.

[20]. S. P. and D. J. Amjad Hussain Bhat, "Machine learning approach for intrusion detection on cloud virtual machines," Int. J. Appl. or Innov. Eng. Manag., vol. 2, no. 6, pp. 57– 66, 2013. [22]

[21]. Z. Salehi, A. Sami, and M. Ghiasi, "Using feature generation from API calls for malware detection," Comput. Fraud Secur., vol. 2014, no. 9, pp. 9–18, 2014, doi: 10.1016/S13613723(14)70531-7.

[22]. S. Yerima, S. Sezer, I. M.-2014 E. international, and undefined 2014, "Android malware detection using parallel machine learning classifiers," ieeexplore.ieee.org, pp. 10–14, 2014, Accessed: Jul. 18, 2022.

[23]. P. V. Shijo and A. Salim, "Integrated static and dynamic analysis for malware detection," Procedia Comput. Sci., vol. 46, pp. 804–811, 2015, doi: 10.1016/J.PROCS.2015.02.149.

[24]. Y. Li, R. Ma, R. J.-I. J. of S. and I. Applications, and undefined 2015, "A hybrid malicious code detection method based on deep learning," covert.io, vol. 9, no. 5, pp. 205–216, 2015, doi: 10.14257/ijsia.2015.9.5.21.

[25]. J. Teknologi et al., "Feature selection and machine learning classification for malware detection," journals.utm.my, vol. 77, pp. 2180–3722, 2015, Accessed: Jul. 18, 2022.

[26]. C. I. Fan, H. W. Hsiao, C. H. Chou, and Y. F. Tseng, "Malware detection systems based on API log data mining," in Proceedings - International Computer Software and Applications Conference, 2015, vol. 3, pp. 255–260. doi: 10.1109/COMPSAC.2015.241.

[27]. Q. Jamil and M. A. Shah, "Analysis of machine learning solutions to detect malware in android," in 2016 6th International Conference on Innovative Computing Technology, INTECH 2016, 2017, pp. 226–232. doi: 10.1109/INTECH.2016.7845073.

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

[29]. W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL 4 MD : A Deep Learning Framework for Intelligent Malware Detection," covert.io, pp. 61–67, 2016, Accessed: Jul. 18, 2022. [Online]. Available: https://www.covert.io/research-papers/deep-learningsecurity/DL4MD- A Deep Learning Framework for Intelligent Malware Detection.pdf

[30]. H. S. Galal, Y. B. Mahdy, and M. A. Atiea, “Behavior-based features model for malware detection,” J. Comput. Virol. Hacking Tech., vol. 12, no. 2, pp. 59–67, May 2016, doi: 10.1007/S11416- 015-0244-0.

[31]. A. Karim, R. Salleh, and M. K. Khan, “SMARTbot: A behavioral analysis framework augmented with machine learning to identify mobile botnet applications,” PLoS One, vol. 11, no. 3, Mar. 2016, doi: 10.1371/JOURNAL.PONE.0150077.

[32]. Y. Cheng, W. Fan, W. Huang, and J. An, “A Shellcode Detection Method Based on Full Native API Sequence and Support Vector Machine,” in IOP Conference Series: Materials Science and Engineering, 2017, vol. 242, no. 1. doi: 10.1088/1757899X/242/1/012124.

[33]. D. Moon, H. Im, I. Kim, and J. H. Park, “DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks,” J. Supercomput., vol. 73, no. 7, pp. 2881–2895, Jul. 2017, doi: 10.1007/s11227-015-1604-8.

[34]. O. Aslan and R. Samet, “A Comprehensive Review on Malware Detection Approaches,” IEEE Access, vol. 8. 2020. doi: 10.1109/ACCESS.2019.2963724.

[35]. R. Mosli, R. Li, B. Yuan, Y. P.-I. I. C. on Digital, and undefined 2017, “A behaviorbased approach for malware detection,” Springer, vol. 511, pp. 187–201, 2017, doi: 10.1007/978-3-319-67208-3_11.

[36]. R. Nix and J. Zhang, “Classification of Android apps and malware using deep neural networks,” in Proceedings of the International Joint Conference on Neural Networks, 2017, vol. 2017-May, pp. 1871–1878. doi: 10.1109/IJCNN.2017.7966078.

[37]. S. Hou, A. Saas, L. Chen, Y. Ye, T. B.-P. of the 2017 IEEE, and U. 2017, “Deep neural networks for automatic android malware detection,” dl.acm.org, pp. 803–810, Jul. 2017, doi: 10.1145/3110025.3116211.

[38]. H. J. Zhu, T. H. Jiang, B. Ma, Z. H. You, W. L. Shi, and L. Cheng, “HEMD: a highly efficient random forest-based malware detection framework for Android,” Neural Comput. Appl., vol. 30, no. 11, pp. 3353–3361, Dec. 2018, doi: 10.1007/s00521-0172914- y.

[39]. P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, “A novel dynamic android malware detection system with ensemble learning,” IEEE Access, vol. 6, pp. 30996–31011, 2018, doi: 10.1109/ACCESS.2018.2844349.

[40]. P. Yan and Z. Yan, “A survey on dynamic mobile malware detection,” Softw. Qual. J., vol. 26, no. 3, pp. 891–919, Sep. 2018, doi: 10.1007/S11219-017-9368-4.

[41]. T. D. Phan and N. Zincir-Heywood, “User identification via neural network based language models,” Int. J. Netw. Manag., vol. 29, no. 3, May 2019, doi: 10.1002/NEM.2049. [43] S. Arshad, M.

[42]. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, “SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System,” IEEE Access, vol. 6, pp. 4321– 4339, 2018, doi: 10.1109/ACCESS.2018.2792941.

[43]. E. M. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, “MalDozer: Automatic framework for android malware detection using deep learning,” DFRWS 2018 EU - Proc. 5th Annu. DFRWS Eur., pp. S48–S59, 2018, doi: 10.1016/J.DIIN.2018.01.007.

[44]. C. Hasegawa and H. Iyatomi, “One-dimensional convolutional neural networks for Android malware detection,” in Proceedings - 2018 IEEE 14th International Colloquium on Signal Processing and its Application, CSPA 2018, 2018, pp. 99–102. doi: 10.1109/CSPA.2018.8368693.

[45]. 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 2018, 2018, vol. 2018-Janua, pp. 1–6. doi: 10.1109/ICCE.2018.8326293.

[46]. S. Naz and D. K. Singh, "Review of Machine Learning Methods for Windows Malware Detection," 2019. doi: 10.1109/ICCCNT45670.2019.8944796.

[47]. K. Sethi, R. Kumar, L. Sethi, P. Bera, and P. K. Patra, "A novel machine learning based malware detection and classification framework," 2019. doi: 10.1109/CyberSecPODS.2019.8885196.

[48]. H. Sayadi et al., "2smart: A two-stage machine learning-based approach for run-time specialized hardware-assisted malware detection," ieeexplore.ieee.org, Accessed: Jul. 19, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8715080/

[49]. A. Mehtab et al., "AdDroid: Rule-Based Machine Learning Framework for Android Malware Analysis," Mob. Networks Appl., vol. 25, no. 1, pp. 180–192, Feb. 2020, doi: 10.1007/S11036-019- 01248-0.

[50]. F. Mercaldo and A. Santone, "Deep learning for image-based mobile malware detection," J. Comput. Virol. Hacking Tech., vol. 16, no. 2, pp. 157–171, Jun. 2020, doi: 10.1007/S11416-019-00346- 7.

[51]. Z. Ma, H. Ge, Z. Wang, Y. Liu, and X. Liu, "Droidetec: Android Malware Detection and Malicious Code Localization through Deep Learning," arXiv Prepr., Feb. 2020, Accessed: Jul. 19, 2022. [Online]. Available: http://arxiv.org/abs/2002.03594

[52]. H. Naeem et al., "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," Ad Hoc Networks, vol. 105, 2020, doi: 10.1016/j.adhoc.2020.102154.

[53]. X. Pei, L. Yu, and S. Tian, "AMalNet: A deep learning framework based on graph convolutional networks for malware detection," Comput. Secur., vol. 93, 2020, doi: 10.1016/j.cose.2020.101792.Z. Cheng, X. Chen, Y. Zhang, S. Li, and Y. Sang, "Detecting Information Theft Based on Mobile Network Flows for Android Users," 2017. doi: 10.1109/NAS.2017.8026853. [56] C. B.-D. Mining and knowledge discovery and undefined 1998, "A tutorial on support vector machines for pattern recognition," Springer, vol. 2, pp. 121–167, 1998, Accessed: Jul. 20, 2022. [Online].