IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 1, 2024

ROBUST INTELLIGENT MALWARE DETECTION USING DEEP LEARNING

Dr.K.Gunasekaran, Professor, Department Of CS SICET, Hyderabad

Illendula Sadhvika, Maddi Srinidhi,Vangari Sri Ranganath, Challa N S Prudhvi Rohith Kumar

UG Student, Department Of CS, SICET, Hyderabad

ABSTRACT

Vulnerabilities caused by malware attacks continue to increase and have become a significant security p roblem in the digital age. Malware detection is still a hot research topic because malware attacks are inc reasing and affecting many computer users, businesses, and governments. Currently, malware is seeking solutions in static and dynamic analysis of malware signatures and behavioral patterns; This is very tim e consuming and has proven to be ineffective at instantly detecting known malware. Recent malware us es polymorphism, morphing, and other evasion methods to rapidly change malware behavior and create a flood of new malware. This new type of malware is often different from existing malware, and machin e learning algorithms (MLA) have recently been adopted for effective malware analysis. However, such processes take a long time because complex architecture requires special training and artistic representa tion. The engineering process can be completely avoided by using advanced MLA methods such as dee p learning. Recently published studies in this direction have shown the performance of their algorithms with biased data, which actually limits the realtime use of their strategies. There is an urgent need to red uce bias and selfevaluate this process to find new ways to detect badly dated malware. To fill this gap in the literature, this paper first evaluates classical MLA and deep learning for the detection, classification and classification of malware using different public and private databases. Second, by making a distinct ion between public and private data and using different time periods to report and evaluate the standard in different ways, we eliminate any data biases removed from the test analysis. Third, our main contribu tion will be to propose a new image processing method with the negative view of MLA and deep learnin g to achieve effective malware detection models. A comparative study of our models shows that our dee p learning architecture outperforms classical MLA. We are delivering the first zeroenergy day in malwa re detection by providing visualization and deep learning for integrated static, dynamic and image proce ssing applications in big data environments. Overall, this article presents a method for effective malware detection using deep learning for rapid deployment. Static and dynamic analysis, artificial intelligence, machine learning, deep learning, image processing, scalable and hybrid frameworks. Introductionn the d igital world of Industry 4.0, rapid advances in technology affect your daily business activities and perso nal life. Internet of Things (IoT) and ApplicationsSenior Researcher Corrado Mencar participated in the review of this text and approved its publication. . However, security issues pose a significant challenge t o realizing the benefits of the Industrial Revolution, with cybercriminals attacking personal computers a nd networks, stealing confidential information to make money, and denying service to machines. These types of attackers use malware or malware to pose a serious threat

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 12, Issue 1, 2024

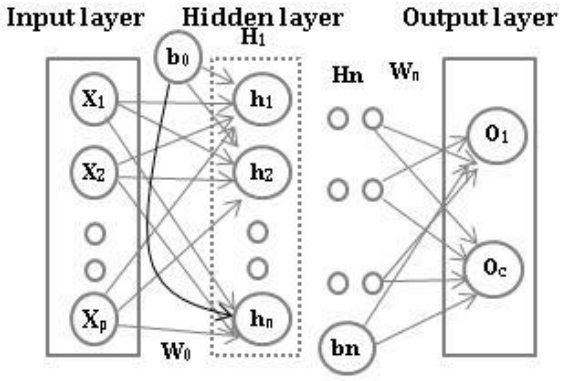
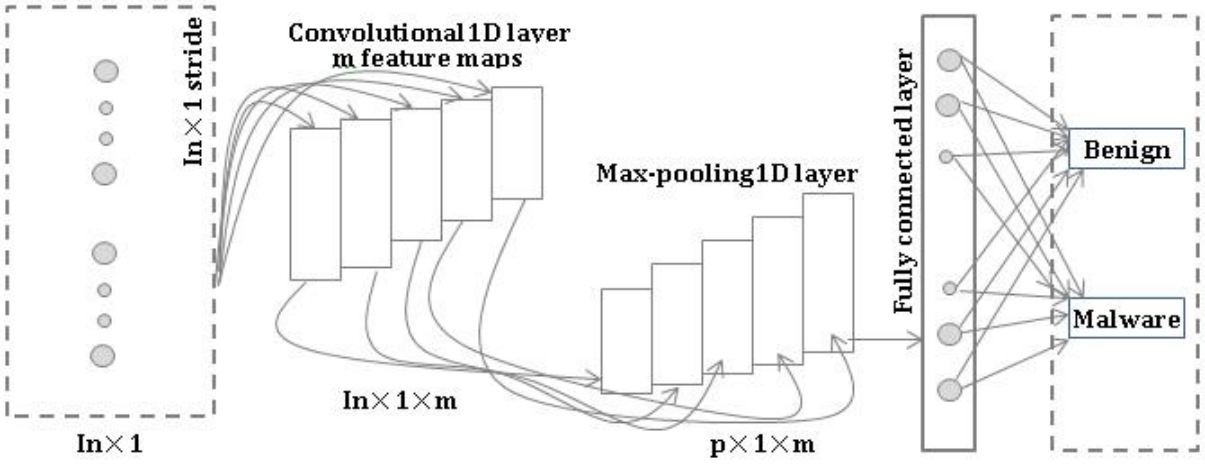


FIGURE 1. Architecture of DNN with n hidden layers.

gradient issue [36]. *ReLU* has been turned out to be more pro-cient and capable of accelerating the entire training process altogether. *ReLU* is de ned mathematically as follows:

$$f(x) \text{ D } \max(0; x) \qquad (2)$$

where x denotes input.



Malicious software classification completely eliminates the need for disassembly and decompilation. re moves, decrypt or execute binaries compared to existing files. Since hyperparameters play an important role in achieving better performance, the reported results can be further improved by looking for vulnera bilities. It is often referred to as big data. Big data brings many challenges to machine learning algorith ms and deep learning due to its characteristics of diversity, speed, and high accuracy. This should includ e details of information seeking and information processing. While autoencoders are the most commonl y used dimensionality reduction methods in deep learning, the most frequently used classical dimension ality reduction methods in classical machine learning are principal component analysis (PCA) and langu

age value decomposition (SVD). Recently, the application of autoencoders has been adopted in network security applications [16]. An autoencoder is a design that learns a latent representation of different obj ects. It learns important features in an unsupervised manner and is seen as a suitable method for analyzi ng network connections as the data generated increases,

TABLE 9. Detailed Data set 1 test results of various classical machine learning classifiers.

Family Name	LR		NB		KNN		DT		AB		RF		SVM rbf		SVM linear	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Adialer.C	1.0	0.0	1.0	0.0	1.0	0.004	1.0	0.0007	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Agent.FYI	1.0	0.0	0.971	0.0	1.0	0.0	0.971	0.0007	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Allaple.A	0.757	0.12	0.618	0.041	0.164	0.0	0.759	0.088	0.995	0.975	0.713	0.055	0.746	0.084	0.811	0.114
Allaple.L	0.688	0.083	0.141	0.933	0.0	0.0	0.689	0.066	0.0	0.0	0.992	0.112	0.937	0.099	0.683	0.069
Alueron.gen!J	0.847	0.003	0.881	0.0	0.034	0.0	0.915	0.005	0.0	0.0	0.915	0.0	0.881	0.0	0.949	0.0
Autorun.K	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0007	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
C2Lop.P	0.167	0.014	0.6	0.013	0.0	0.0	0.133	0.015	0.0	0.0	0.4	0.013	**0.467**	**0.014**	**0.45**	**0.01**
C2Lop.gen!G	0.091	0.009	0.432	0.013	0.0	0.0	0.25	0.016	0.0	0.0	0.068	0.002	0.091	0.003	0.272	0.008
Dialplatform.B	0.962	0.0	0.962	0.0	0.962	0.0	0.962	0.001	0.0	0.0	0.962	0.0	0.962	0.0	0.962	0.0
Dontovo.A	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Fakerean	0.982	0.001	0.965	0.0	0.947	0.0	0.982	0.003	0.0	0.0	0.982	0.0	0.982	0.0	0.982	0.0
Instantaccess	1.0	0.0	0.977	0.0	1.0	0.0	1.0	0.002	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Lolyda.AA 1	0.906	0.002	0.875	0.0004	0.844	0.0	1.0	0.0007	0.0	0.0	0.922	0.0004	0.922	0.0	0.922	0.0
Lolyda.AA 2	0.982	0.0004	0.855	0.001	0.927	0.0	1.0	0.0	0.0	0.0	1.0	0.0007	1.0	0.0	1.0	0.0004
Lolyda.AA 3	1.0	0.002	0.946	0.0	1.0	0.586	1.0	0.003	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0004
Lolyda.AT	0.833	0.002	1.0	0.005	0.021	0.0	0.875	0.003	0.0	0.0	0.938	0.0004	**0.958**	**0.0**	**1.0**	**0.001**
Malex.gen!J	0.439	0.006	0.61	0.0004	0.0	0.0	0.854	0.003	0.0	0.0	0.829	0.0	0.0	0.0	0.415	0.0025
Obfuscator.AD	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.003	1.0	0.0	1.0	0.0	1.0	0.0
Rbot!gen	0.872	0.003	0.957	0.0	0.489	0.0	0.9149	0.005	0.0	0.0	1.0	0.001	1.0	0.0	1.0	0.0007
Skintrim.N	0.917	0.0	1.0	0.001	0.708	0.0	0.75	0.001	0.0	0.0	0.917	0.0	0.917	0.0	1.0	0.0
Swizzor.gen!E	0.263	0.009	0.477	0.008	0.0	0.0	0.132	0.01	0.0	0.0	0.004	0.237	0.237	0.003	0.342	0.006
Swizzor.gen!I	0.15	0.007	0.35	0.008	0.0	0.0	0.175	0.012	0.0	0.0	0.225	0.0029	**0.25**	**0.0025**	**0.375**	**0.006**
VB.AT	0.861	0.004	0.869	0.0008	0.844	0.0	0.803	0.002	0.0	0.0	0.902	0.002	**0.992**	**0.001**	**0.967**	**0.0003**
Wintrim.BX	0.687	0.0004	0.552	0.0	0.517	0.0	0.758	0.001	0.0	0.0	0.793	0.0	0.655	0.0	0.689	0.0
Yuner.A	1.0	0.0004	1.0	0.0	1.0	0.0	1.0	0.0004	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Accuracy(%)	**78.6**		**80.5**		**41.8**		**79.5**		**33.0**		**84.3**		**83.7**		**82.8**	

TABLE 10. Confusion matrix for CNN 2 C LSTM architectur

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 12, Issue 1, 2024

Malware Family	Adialer.C	Agent.FYI	Allaple.A	Allaple.L	Alueron.gen!J	Autorun.K	C2Lop.P	C2Lop.gen!G	Dialplatform.B	Dontovo.A	Fakerean	Instantaccess	Lolyda.AA 1	Lolyda.AA 2	Lolyda.AA 3	Lolyda.AT	Malex.gen!J	Obfuscator.AD	Rbot!gen	Skintrim.N	Swizzor.gen!E	Swizzor.gen!I	VB.AT	Wintrim.BX	Yuner.A	Error rate (%)
Adialer.C	**37**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Agent.FYI	0	**35**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Allaple.A	0	0	**871**	8	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0	1.582
Allaple.L	0	0	2	**475**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.419
Alueron.gen!J	0	0	0	0	**59**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Autorun.K	0	0	0	0	0	**32**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
C2Lop.P	0	0	2	0	0	0	**41**	5	0	0	1	1	0	0	0	0	1	0	0	0	5	2	0	2	0	31.667
C2Lop.gen!G	0	0	1	0	0	0	9	**33**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	25
Dialplatform.B	0	0	0	0	0	0	0	0	**52**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1.887
Dontovo.A	0	0	0	0	0	0	0	0	0	**49**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Fakerean	0	0	0	0	0	0	0	0	0	0	**114**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Instantaccess	0	0	0	0	0	0	0	0	0	0	0	**129**	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Lolyda.AA 1	0	0	1	0	0	0	0	0	0	0	0	0	**63**	0	0	0	0	0	0	0	0	0	0	0	0	1.563
Lolyda.AA 2	0	0	0	0	0	0	0	0	0	0	0	0	0	**55**	0	0	0	0	0	0	0	0	0	0	0	**0**
Lolyda.AA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**37**	0	0	0	0	0	0	0	0	0	0	**0**
Lolyda.AT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**48**	0	0	0	0	0	0	0	0	0	**0**
Malex.gen!J	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	**36**	0	0	0	0	0	0	0	0	12.195
Obfuscator.AD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**43**	0	0	0	0	0	0	0	**0**
Rbot!gen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**47**	0	0	0	0	0	0	**0**
Skintrim.N	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**23**	0	0	0	0	0	4.167
Swizzor.gen!E	0	0	0	0	0	0	8	1	0	0	0	0	0	0	0	0	0	0	0	0	**14**	14	0	1	0	63.158
Swizzor.gen!I	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	11	**21**	0	0	0	47.5
VB.AT	0	0	1	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	**118**	0	0	3.279
Wintrim.BX	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**25**	0	13.793
Yuner.A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**240**	**0**

IX CONCLUSION

This paper evaluated classical machine learning algorithms (MLA) and deep learning architectures based on static analysis, dynamic analysis, and image processing techniques for malware detection and proposed a highly scalable framework called ScaleMalNet for detecting, classifying, and categorizing zero-day malware. This framework applies deep learning to collected malware from end-user hosts and follows a two-stage malware analysis process. In the first phase, a hybrid of static and dynamic analysis was used to classify the malware. In the second stage, the malware was grouped into corresponding malware categories using image processing techniques. Various experimental analyzes performed by applying variation in models to both publicly available benchmark datasets and privately collected datasets in this study showed that deep learning-based methodologies outperformed classical MLA. The developed framework is capable of analyzing a large number of malwares in real-time and is scaled to analyze an even larger number of malwares by stacking several additional layers on existing architectures. Future research includes

ISSN 2347–3657
Volume 12, Issue 1, 2024

exploring these variations with new features that could be added to existing data. The main implications of this work, weaknesses and future scope can be summarized as follows:A two-stage process-scalable malware detection framework is proposed. It uses the proposed framework4673state-of-the-art method, deep learning, which detects malware at the first level and at the second level, the malware is categorized into the corresponding categories.The performances obtained by deep learning architectures outperformed classical MLA in static, dynamic and image processing-based malware detection and cat-egorization. However, in the study of dynamic analysis-based malware detection, deep learning architectures are applied to features obtained from domain knowledge. This can be avoided by collecting memory dumps for binaries at runtime and then mapping the memory dump file to a grayscale image.In image processing using a deep learning-based malware identification study; malware were transformed into xed-sized images and then tracked. In future work, the Spatial Pyramid Pooling (SPP) layer can be used to allow images of any size as input. This learns features at different scales and can be placed between a subsampling layer and a fully connected layer to improve the flexibility of our models.Malware families in the Malimg dataset are highly unbalanced. A cost-sensitive approach can be used to address the imbalance problem of multi-class malware families. This makes it easy to introduce cost items into the back-learning methodology of deep learning architectures. Primarily, the cost item represents the importance of the classification, which provides a lower value for classes with a large number of samples and a higher value for classes with a smaller number of samples.Deep learning architectures are vulnerable in hostile environments [16]. The generative adversarial network method can be used to generate samples during the testing or deployment phase that deep learning architectures can easily fool. The robustness of deep learning architectures is not discussed in the proposed work. This is one important direction for future work, as malware defection is an important application in security-critical environments. A single misclassification can cause several damages to an organization.

CONFIRMATION

The authors would like to thank NVIDIA India for the GPU hardware support for the research grant. They would also like to thank the Computational Engineering and Networking (CEN) department for research support.

REFERENCES

[1]. "Machine Learning Techniques in Cybersecurity." Encyclopedia. Retrieved from https://encyclopedia.pub/entry/25675.
[2]. Abdullah, A. H., Ahmed, M. H., & Wahab, M. H. A. (2021). A Comparative Study of Network Intrusion Detection Techniques Using NSL-KDD Dataset. IEEE Access, 9, 91924-91942.
[3]. Akhtar, S., Faisal, M., Ahmad, S., & Rho, S. (2020). Machine learning-based ransomware detection: State-of-the-art and future research directions. Journal of Network and Computer Applications, 153, 102539.
[4]. Akinyele, J. R., Gao, K., & Zhu, S. (2015). Insider threat detection using log analysis and machine learning. International Journal of Information Security, 14(5), 403-415.

ISSN 2347–3657
Volume 12, Issue 1, 2024

[5]. Alawami, A. K., Khan, M. K., & Kiong, T. E. (2020). Insider threat detection: A review and research directions. Journal of Network and Computer Applications, 153, 102531.
[6]. Alazab, M., Hobbs, M., & Abawajy, J. (2018). A survey of botnet detection techniques. Journal of Network and Computer Applications, 110, 60-71.
[7]. Alzahrani, B., Zulkernine, M., & Alazab, M. (2020). Machine learning-based intrusion detection techniques for securing industrial control systems: A review. Computers & Security, 88, 101628.
[8]. Bhattacharya, S., Gupta, P., & Chatterjee, J. (2021). A comparative study of machine learning algorithms for malware detection. Multimedia Tools and Applications, 80(10), 14935-14957.
[9]. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 15.
[10]. Chiong, R., Lee, V. C., & Zhou, L. (2017). Anomaly detection in cyber security: A machine learning approach. In Machine learning paradigms: Advances in data analytics (pp. 81-112). Springer, Cham.
[11]. Demertzis, K., & Karampelas, P. (2020). A review of anomaly detection techniques in financial markets: An application to emerging markets. Expert Systems with Applications, 146, 113172.
[12]. Dhamecha, T. I., & Thakkar, P. (2020). A Comprehensive Review on Anomaly Detection Techniques using Machine Learning. International Journal of Advanced Research in Computer Science, 11(4), 44-51.
[13]. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2018). Data augmentation using synthetic data for time series classification with deep residual networks. arXiv preprint arXiv:1808.08467.