De e op en nd n e on o c ne e n n nd A p e n eco n on n e de ec on n e nd e n y o ode

Madeleine Rabitoy

Davenport University

Summer 2023

In fulfillment of the requirements for the Degree of Master of Science in Information Assurance and Cyber Security

e o Con en

A c

Malware detection is a field that is constantly in flux. Increasingly organized and intelligent malware distributors and authors make the detection and prevention of malware infections more difficult now than it has ever been before, as does the proliferation of easy-to-use malware creation tools and more complex and dangerous samples. Modern warfare and international cybercrime has also increasingly involved cyberattacks, targeted distribution attacks and highly

C pe L. e e e .e

This literature review is intended to contribute the following:

- Present a summary of the current state of the field of machine learning-powered malware detection and prevention.
- Identify current trends in malware detection pattern recognition and new approaches to the detection of novel malware.
- Discuss the challenges, limitations and gaps in knowledge that are presented by the current state of machine learning integration into malware detection.
- Evaluate potential avenues for future research and study that have not yet been explored in the known literature on this subject.

ynt

Machine learning is a constantly growing, rapidly evolving subfield of artificial intelligence that involves the use of algorithms and statistical models to enable a system to improve its performance on a specific, narrow task over a period of time and through many cycles of trial and error. The training process for a machine learning system is typically done by feeding the system a large amount of training data and allowing it to define correlations, identify meaningful patterns and make useful predictions or mimic a specific desired behavior based on those patterns, after which these patterns and predictions are reinforced and guided by a human overseer to encourage or discourage specific interpretations and improve performance on edge cases. Machine learning has a wide range of potential and practical applications in the field of computer science, and has already been applied to a wide array of tasks such as image generation and recognition, sentiment analysis, conversational text generation, facial recognition, natural language processing, and predictive modeling.

There is a great deal of existing literature on the integration of machine learning in malware detection, most of it from the past ten years as the field of machine learning overall has rapidly developed and flourished. Most of this literature acknowledges the assertion that signature-based analysis is limited by its database of known signatures, and will not detect novel or unknown samples of malware. The literature also recognizes that these databases must be constantly updated in order to remain useful in a rapidly changing and evolving malware development landscape, and as such they require a great deal of manual maintenance and upkeep by cybersecurity experts who will be constantly obligated to survey malware landscapes and keep up with “industry” trends. Finally, the literature notes that signature-based analysis will not identify variants or zero-day exploits, as both of these will not be available in the signature database ahead of time for the system to be aware of their presence. Machine learning methods have the potential to revolutionize the field of malware detection with the ability to detect patterns rather than individual signatures, expanding the capability of these systems with regard to detecting novel samples and lessening the workload required to maintain these systems once they reach a certain level of complexity. However, almost every paper studied in the literature review noted limitations, challenges and daunting knowledge gaps in the idea of fully integrating machine learning into the problem of malware detection to produce a more effective and robust detection system, as described further below.

L t t on nd n

Most of the literature identifies several key limit intensive and demanding process for most computers, and as such the applications of advanced machine learning algorithms may not be feasible for home and personal computers with limited computing power or tolerance for latency. The literature also notes that machine learning training databases are often very large and will take up a cumbersome amount of storage space on most machines, limiting their practicality on older machines or machines with less disk space, and the quick and efficient retrieval of this data for detection purposes may not be possible anymore as the database grows larger and larger and searches become more computationally difficult. As noted by several papers in the review, additional layers of complexity and protection will result in additional performance problems, and trying to add further rules and edge cases to account for more and more exotic strains of malware only complicates this further.

Another limitation of this approach noted in the literature is that real-time detection is difficult to reconcile with current machine learning algorithm execution speeds. The real-time detection and prevention of malware is one of the most critical components of a modern antivirus, and many modern antivirus programs are able to block malware programs from executing in real time before they even cause damage and promptly alert the user of the malicious program's presence. This capability is incredibly important for the function of a useful antivirus, and classical machine learning algorithms often cannot produce these split-second results fast enough to prevent malware from doing damage once it is detected. By the time the artificial intelligence is aware of it, the malware may already have begun to do damage or execute its malicious payload, which is an undesirable outcome for the end user. Home versions of antivirus software are often expected to be responsive to the user and relatively fast to react to their actions, and machine learning is traditionally not a reactive, user-friendly or responsive design space. Various strategies have been attempted in the literature to reconcile this problem, such as splitting malware machine learning algorithms into speed categories and attempting faster detection approaches that don't rely on traditional file-based detection.

A few of the papers studied in the literature attempt to explore and refute this limitation, such as J. Saxe and K. Berlin (2015)[16], whose conclusions contend that a small, accurate and effective machine learning system can indeed run in real time on a real environment. Other papers,

y r d r M hods

J. Saxe and K. Berlin (2015)[16] used a hybrid machine learning approach that collectively incorporated contextual byte features, import patterns, and a calibration-based scoring model. They were able to achieve a 95% accuracy rate with their deep learning system using this methodology, with a 0.1% false positive rate (FPR) based on over 400,000 malware samples and malicious binaries.[16] They concluded that their results were a promising indicator that it was now possible and feasible for everyday customers to run a small, robust and accurate machine

for machine learning systems. They used a common mathematical formula known as the nearest-neighbor algorithm to determine how similar an unknown, potentially novel malware n-gram was to known malicious malware n-grams, with high neighbor scores resulting in a higher detection score. The next logical step would then be to procedurally generate code samples that were neighbor-scored as close to other known samples, such that they were also malicious but were unknown to current signature databases. A large enough corpus of this type could potentially be able to recognize malicious n-grams inside of novel malware samples and flag the samples, given that there are only so many ways to code a given malicious function that are highly dissimilar.[15] The results of this experiment produced a database of 2,000 file signatures made up of malicious n-grams, with a detection rate of 69.66% for 2-grams and a maximum detection rate of 91.25% for 4-grams.[15]

t y

Many survey papers have been written about machine learning and statistical approaches to malware detection. The surveys that were located and synthesized in this literature review are noted below and described in detail. Several of them also proved to be intellectually valuable for my research project, as discussed further below.

Y M n h n L n nd Y

This paper served as a comprehensive literature review of the field overall and summarized some of the same issues and challenges that I described above in the synthesis. It also discussed a few potential research directions that the field had yet to explore. It mentioned how the proliferation of polymorphic and metamorphic malware had resulted in significantly more challenges in malware detection in the past decade, as well as how research in this area had advanced to a stage where a more thorough review of the literature was needed to address these new challenges. The authors generally assert that the identification of malware through malicious features and behaviors, using machine learning as an asset to this process, will inevitably eclipse signature-based detection methods. The survey was presented to the 2021 International Conference on Computer Information Science and Artificial Intelligence in Kunming, China.

yy h n B r d M h n A L Y

This paper was a survey of recent deep learning trends in machine-based malware detection specifically, and focused on performance limitations, conventional and modern machine learning technique comparisons, and statistical analysis of the methods commonly used in the literature. They also discussed more thoroughly the latency problems and network limitations that machine learning introduces into malware detection systems, and suggest that large systems may need to be broken up into smaller modules and subsystems to address these challenges, with smaller subsets of the training data appropriate for each module’s function. This is an interesting approach that I feel deserves more attention and discussion – perhaps my research could explore the idea of training a system specifically on ransomware, or specifically on remote access

Trojans, and so on. The survey was presented this year at the Pakistan Institute of Engineering and Applied Sciences in Nilore, Pakistan.

chh o h sh nd B B

This paper was a study of malware classification techniques with machine learning specifically and had a narrow focus on how the field of cybersecurity classifies various types of malware, and how those types translate to machine learning labels. It discussed how many modern antiviruses assign malware into families, such as the WannaCry family of ransomware or the Cryxos family of Trojans, and how these labels can be both useful and too generalized to be helpful. It was an interesting discussion mainly because what initially seems like a very easy problem – tell what kind of malware you have – is actually far more complicated than it seems, especially because malware authors have an incentive to obfuscate that information or combine various types of malware together to evade detection and increase the sample’s reach. The problem whath

C pe 3 e e c Me od

nt n

For the experimental component of this survey project, I will be examining real-world sample data and testing real machine learning models on various curated research datasets that were uncovered during the literature review; these datasets and their origins, contents and sources are described in more detail further below. This experimental process will entail the examination, incorporation and testing of machine learning models that other authors have created in past surveys of this kind, in order to determine whether older machine learning models can still perform reasonably well on modern malware samples.

Many of the previous experimental papers in this field resulted in the creation of testable machine learning models that performed well on specific subsets of malware data, such as Zhu et. al (2017)[10] and their “DeepFlow” machine learning model. The main methodology of this portion of the project will be gaining access to these models, wherever possible, and testing them on various types of data beyond the scope in which the original authors tested them, as well as adjusting them for modern and exotic types of malware to examine the impact on their performance and accuracy in order to challenge my hypothesis that machine learning improves the outcomes of these detection runs. These models will serve as real-world demonstrations of my experimental thesis that the field of malware detection 8( )-0.4794-0.-550.4758(h)-0.953971(a)3.157857028(

from a single metric. A high F1 score indicates that a model has good precision and recall, while a lower F1 score typically indicates that a model is imperfect in one or both of these areas.

s os [illegible] [illegible]

A fundamental part of evaluating a malware detectio

- **e d e** consisting of several thousand modern, curated malware samples that have been verified as malicious samples by VirusTotal.
- **eZoo d e** consisting of several thousand modern and well-known samples curated on GitHub.

C p e e

t t Ac t on

The first step in my experimental process was to attempt to acquire access to the datasets described above and decide which ones would be the most directly useful for my research purposes, by examining each database collated from the literature review and determining the merits and drawbacks of incorporating each one into my research. A major hurdle that I encountered was the fact that several of these databases are over a decade old, meaning that all of their malware samples are only functional on older machines and operating systems that are no longer supported or considered secure. Many databases that I examined were also no longer actively supported or were locked behind verification, limited access, dead links, web archives, or other barriers to access. Furthermore, I encountered other databases during this stage of my research that I had not previously discovered as part of the literature review.

h r s h r s

In the process of this stage of my research investigation, I uncovered a malware database that I found extremely useful and had not discovered in my preliminary research. The VirusShare database of malware and hashes proved to be a highly valuable collection of actively traded, verifiably malicious, modern malware samples collated and curated by VirusTotal analyses, and it has been used in a great deal of recent publications and research, including Abbasi et. al. (2020)[41]. The database of hash sets is publicly available, while individual samples required verification to download for research purposes. I acquired access to these large hash set files through a refined version provided by MantaRay Forensics, and curated and prepared these files such that they would be easy to ingest by my machine learning models, which proved extremely valuable for testing.

h oo

Another malware source that was uncovered after the literature review was The Zoo, a live malware repository hosted on Github that allows the study of live malware samples. This valuable trove of malware information included live samples of infamous and well-known ransomware such as WannaCry, Jigsaw, CryptoLocker, Zeus, and TeslaCrypt, which I tested by successfully infecting a virtual machine environment running Windows 8 and another virtual machine running Windows 10.

Mod Ac t on

After acquiring access to the necessary malware and file hash datasets and formatting them for model ingestion, which included devising a few custom Python scripts for feature extraction, I next sought to obtain access to some of the machine learning-based malware classification models that were uncovered as part of the literature review and further investigations. The models I was able to obtain access to for testing and research use are described below.

- *ro* The **L BM e NN ndo o e Mode** and **L ne M C e** models featured in the paper “Explanation-Guided Backdoor Poisoning

Attacks Against Malware Classifiers" by Severi, Meyer, Coull & Oprea[42]. The goal of the authors was to demonstrate the weaknesses of a variety of machine learning models against a specific type of backdoor poisoning attack where a machine learning dataset is corrupted by malicious attackers, but it featured a variety of malware classifier models as part of their experiment. The models are conveniently hosted on Github (github.com/ClonedOne/MalwareBackdoors).

- *ro* The **dened DNN Mode** featured in the paper "Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware Detection" by D. Li & Q. Li[43]. The code is available on Github (github.com/deqangss/adv-dnn-ens-malware).
- *ro* The **Con o on Ne Ne o CNN Mode** (Kyadige and Rudd et al., https://arxiv.org/abs/1905.06987) **e " d en Boo ed Dec on ee BD Mode** (Anderson and Roth, https://arxiv.org/abs/1804.04637), and **M Con " By e Le e CNN** (Raff et al., https://arxiv.org/abs/1710.09435), featured in the paper "Quo Vadis: Hybrid Machine Learning Meta-Model Based on Contextual and Behavioral Malware Representations" by Trizna, Dmitrijs (2022)[44]. The model code is provided on Github (github.com/dtrizna/quo.vadis).
- *ro* The **DNN Mode** featured in the paper "Adversarial Deep Learning for Robust Detection of Binary Encoded Malware" by A. Al-Dujaili et al. (2018)[45]. The model code is provided on Github (github.com/ALFA-group/robust-adv-malware-detection).

Of these, Group 1's model set proved to be the most useful for testing, the most programmatically diverse in terms of model variety, and the most readily able to adapt for other, more customized datasets and specific sample files beyond the ones on which it had been tested in the paper. In addition, Group 2's model relied on a deprecated version of TensorFlow that is no longer readily available or compatible with modern packages, and so was not able to be used for testing on the machines and environments I had readily available. Thus, the Group 1 model set was the one that I primarily used for the final testing runs.

Mod t n

At this stage I was finally prepared to test the chosen subset of machine learning models on the datasets that I had acquired and curated. This testing was performed in a normal Windows 10 environment as well as within a Windows 10 virtual machine environment created with VirtualBox, and primarily utilized Python 3.6-3.8, Visual C++ 15, TensorFlow 1.0 (for backwards compatibility functions) and 2.0, SKLearn, and a variety of Python libraries and packages, including Numpy, Joblib and PEFile.

Mod o nc t

The model performance results that were acquired from the testing runs performed above are quantified and summarized below.

L h BM

This model produced the following results on the databases on which it was tested:

MB t / o t / t

	Precision	Recall	F1
	0.83483	0.99833	0.90929
	0.99746	0.76818	0.86793
Acc cy			**9**
M c o A e e	0.91615	0.88326	0.88861
e ed A e e	0.90965	0.89245	0.89026

ont o t t

	Precision	Recall	F1
	0.99750	0.99900	0.99825
	0.99900	0.99750	0.99825
Acc cy			**99**
M c o A e e	0.99825	0.99825	0.99825
e ed A e e	0.99825	0.99825	0.99825

c c / n n d n MB

	Sha256 Hash	Identified?
n o e	3ae96f73d805e1d3995253db4d910300d8442ea603737a1428b	

o C ed

[1] A. Azmoodeh, A. Dehghantanha, M. Conti and K.-K.-R. Choo (2018). "Detecting crypto-ransomware in IoT networks based on energy consumption footprint", J. Ambient Intell. Hum. Comput., vol. 9, no. 4, pp. 1141-1152, Aug. 2018. Retrieved April 2023 from https://www.researchgate.net/publication/319252402_Detecting_crypto-ransomware_in_IoT_networks_based_on_energy_consumption_footprint

[2] A. bin Asad, R. Mansur, S. Zawad, N. Evan and M. I. Hossain (2020). "Analysis of Malware Prediction Based on Infection Rate Using Machine Learning Techniques," 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 706-709, doi:

10.1109/ICISCE48695.2019.00014. Retrieved November 2022 from https://ieeexplore.ieee.org/document/9107644

[19] L. Martignoni, R. Paleari and D. Bruschi (2009). "A framework for behavior-based malware analysis in the cloud", Proc. Int. Conf. Inf. Syst. Secur., 2009.

[20] L. Xiao, Y. Li, X. Huang and X. Du (2017). "Cloud–based malware detection game for mobile devices with offloading", IEEE Trans. Mobile Comput., vol. 16, no. 10, pp. 2742-2750, Oct. 2017.

[21] M. G. Schultz, E. Eskin, F. Zadok and S. J. Stolfo (2001). "Data mining methods for detection of new malicious executables", Proc. IEEE Symp. Secur. Privacy, May 2001.

[22] M. Yeo et al. (2018). "Flow-based malware detection using convolutional neural network," 2018 International Conference on Information Networking (ICOIN), 2018, pp. 910-913, doi: 10.1109/ICOIN.2018.8343255. Retrieved November 2022 from https://ieeexplore.ieee.org/document/8343255

[23] N. B. Akhuseyinoglu and K. Akhuseyinoglu (2016). "AntiWare: An automated Android malware detection tool based on machine learning approach and official market metadata," 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2016, pp. 1-7, doi: 10.1109/UEMCON.2016.7777867. Retrieved November 2022 from https://ieeexplore.ieee.org/document/7777867

[24] N. Nissim, A. Cohen and Y. Elovici (2017). "ALDOCX: Detection of Unknown Malicious Microsoft Office Documents Using Designated Active Learning Methods Based on New Structural Feature Extraction Methodology," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 3, pp. 631-646, March 2017, doi: 10.1109/TIFS.2016.2631905. Retrieved November 2022 from https://ieeexplore.ieee.org/document/7762928

[25] N. Pachhala, S. Jothilakshmi and B. P. Battula (2021). "A Comprehensive Survey on Identification of Malware Types and Malware Classification Using Machine Learning Techniques," 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021, pp. 1207-1214, doi: 10.1109/ICOSEC51865.2021.9591763. Retrieved November 2022 from https://ieeexplore.ieee.org/document/9591763

[26] P. R. K. Varma, K. P. Raj and K. V. S. Raju (2017). "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 294-299, doi: 10.1109/I-SMAC.2017.8058358. Retrieved November 2022 from https://ieeexplore.ieee.org/document/8058358

[27] S. Naz and D. K. Singh (2019). "Review of Machine Learning Methods for Windows Malware Detection," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944796. Retrieved November 2022 from https://ieeexplore.ieee.org/document/8944796

[28] S. Poudyal, K. P. Subedi and D. Dasgupta (2018). "A Framework for Analyzing Ransomware using Machine Learning," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1692-1699, doi: 10.1109/SSCI.2018.8628743. Retrieved November 2022 from https://ieeexplore.ieee.org/document/8628743

[29] S. Poudyal and D. Dasgupta (2020). "AI-Powered Ransomware Detection Framework," 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, pp. 1154-1161, doi: 10.1109/SSCI47803.2020.9308387. Retrieved November 2022 from https://ieeexplore.ieee.org/document/9308387

[30] S. Poudyal and D. Dasgupta (2021). "Analysis of Crypto-Ransomware Using ML-Based Multi-Level Profiling," in IEEE Access, vol. 9, pp. 122532-122547, 2021, doi: 10.1109/ACCESS.2021.3109260. Retrieved November 2022 from https://ieeexplore.ieee.org/document/9526633

[31] S. Poudyal, Z. Akhtar, D. Dasgupta and K. D. Gupta (2019). "Malware Analytics: Review of Data Mining, Machine Learning and Big Data Perspectives," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019, pp. 649-656, doi: 10.1109/SSCI44817.2019.9002996. Retrieved November 2022 from https://ieeexplore.ieee.org/document/9002996

[32] S. Vanjire and M. Lakshmi (2021). "Behavior-Based Malware Detection System Approach For Mobile Security Using Machine Learning," 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), 2021, pp. 1-4, doi:

Appendix A