Survey of Malware Detection through Use of Neural Network Models

Joshua Bean*, Stephen A. Torri†
Department of Computer Science & Engineering
Mississippi State University
jab1896@msstate.edu*, storri@cse.msstate.edu†

2024 International Conference on Cyber-Physical Social Intelligence (ICCSI) | 979-8-3503-7673-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCSI62669.2024.10799495

***Abstract*—In general, antivirus software has heavily relied on signature-based detection methods. These antivirus products usually reference a database of known malicious signatures, making them naturally susceptible to previously unseen malware samples. One of the easiest ways for malware authors to produce these new malware samples is by applying various obfuscation techniques to the previously existing malware. This paper will survey a newer approach to malware detection through neural networks by gathering results from recent journal articles and conference papers published in various online databases. These artificial intelligence-oriented solutions show promise in resilience to unfamiliar malware variants. Explicitly focusing on obfuscated malware, it will compare detection rates between signature-based and neural network-based malware detection systems and organize results using various NN models and training features. Overall, the survey finds that signature-based methods detect between 50 and 80 percent of obfuscated malware samples, while all but one of the neural network methods detect between 90 and 99 percent. This considerable increase in performance suggests that neural networks might provide an effective alternate or supplemental detection method for antivirus products.**

***Index Terms*—neural networks, malware detection, obfuscation, artificial intelligence, antivirus**

I. Introduction

The word “malware” is a combination of the words malicious (mal-) and software (-ware) and is used as a broad category to refer to any unwanted or ill-intending computer program. Malware has existed for almost as long as computers have. Security professionals and malware authors constantly battle as they invent new ways to hide or disguise their malware. The traditional approach to malware detection has been through signature-based detection, usually employing some hashing algorithm and a signature database. The biggest issue with this approach, which malware authors have been keen to exploit, is its vulnerability to obfuscation. In short, creating several obfuscated program versions that look entirely different on a binary level and, therefore, have different signatures is simple.

There are several different methods for obfuscation, which are well-summarized by Elsersy et al. [1]. There are two main categories of obfuscation: polymorphism and metamorphism. Polymorphic code contains a primary payload packed or encrypted with a variable key and decrypts upon execution. Metamorphic malware is a much larger category that includes more subtle transformations such as code reordering, dead code insertion, and other techniques that preserve the semantics of the execution but result in new signatures on each generation.

As artificial intelligence has continued to improve, significant research has been done to apply this field to malware analysis. The flexibility offered by AI models has vast potential to provide a better solution than signature-based detection for obfuscated malware. As explained by Gibert et al. [2], deep learning and neural networks are at the forefront of AI research and have the potential to offer greater freedom than other machine learning algorithms when it comes to input parameters for malware detection. This survey will focus on deep learning and its performance in malware detection compared to signature-based methods. It will also seek to identify the most promising training features for maximizing the accuracy of these neural networks.

Section II will discuss the survey design and objectives. Section III will present the survey results and their implications. Finally, Section IV concludes the survey and suggests future work based on gaps in existing literature.

II. Research Design and Method

The procedures followed by this survey are summarized in the following subsections to ensure replicability.

A. Planning the Review

1. Research Question:
When attempting to detect obfuscated malware (P), how do neural networks trained on various static or dynamic features (I) affect detection rates (O) in comparison to traditional signature-based methods (C)?

2. Inclusion/Exclusion Criteria:
The following outlines the criteria used to select literature for the survey.

Paper is included if:

- It was a peer-reviewed conference paper or journal article.
- It was written in English.
- It was published after 2014 (NN-focused literature). Publications on signature-based methods may be older.
- It contained results relevant to the research question.
- It included numerical detection rate results for neural network systems, or it contained either numerical or qualitative results for signature-based systems.

Paper is excluded if:

- It did not address either detection rates in signature-base or neural network detection of obfuscated malware.
- It was published before 2014 (for NN-focused literature).
- It did not cite an open-source or comprehensive dataset of obfuscated malware (for NN-focused literature).
- It did not cite a transparent malware dataset (signature-focused literature).

3. Data Collection and Analysis
The following is a list of the data points collected from each study to answer the research question.

- Malware target (Android, PE, IoT)
- Malware source
- obfuscation type/tool
- Neural Network type (if applicable)
- Training Features (if applicable)
- Detection Rate

B. Conducting the Review

This survey primarily gathered literature from IEEE Xplore and ACM Digital Library, as well as other online databases. The goal was to find studies that included numerical detection rates so that this survey could provide concrete numbers to answer the research question. The searches were targeted by using different combinations of the following keywords:

- Malware
- Obfuscation
- Neural Network or RNN or CNN or ANN
- Hash-based Detection or Signature-based Detection

One of the challenges that arose in conducting the research was ensuring the relevancy of the studies. In particular, despite the "obfuscation" keyword in the searches, several studies initially found on neural networks did not perform their experiments on an obfuscated malware dataset, which was not immediately apparent from the abstracts of these papers, as some of them even mentioned malware obfuscation, but ultimately did not further address this issue in the content of the study.

III. Results, Discussion, and Implications

Table I contains the list of studies testing signature-based detection methods against obfuscated malware. Table II lists the studies that detail detection rates of various neural networks trained against obfuscated malware. Both tables are sorted by malware type, listing Android-focused and PE-focused studies.

A. Signature-based Detection Results

Examining the signature-based detection results, we see many datasets and obfuscation tools used to perform the various studies. We do, however, see the "Drebin Project" and the "Contagio" Android malware datasets referenced a couple of times by Nawaz et al., Chua et al., and Canfora et al., though with different obfuscation tools used [4] [5] [6].

TABLE I
Signature-based Detection

Study	Malware Target	Dataset Source	Obfuscation Tools	Detection Rate	Conclusion
Hammad et al. [3]	Android	AndroZoo, Android Malware Genome, Contagio, etc.	Allatori, ProGuard, ADAM, DroidChameleon, DashO, Apktool, Jarsigner	67%	obfuscation decreases detection rate
Nawaz et al. [4]	Android	Drebin Project	Apktool	Mostly between 60% and 80% (depends on obfuscation applied)	Obfuscation decreases detection rate
Chua et al. [5]	Android	Drebin Project, Contagio	Custom	51.3%	Obfuscation decreases detection rate
Canfora et al. [6]	Android	Drebin Project	Apktool	All but three antivirus products detect fewer samples post-obfuscation	Obfuscation decreases detection rate
Christodorescu et al. [7]	PE	Custom	Custom	False negative rate over 50% in almost all cases	Obfuscation decreases detection rate
Wong et al. [8]	PE	Assorted sources	NGVCK, G2, VCL32, MPCGEN metamorphic virus generators	Average of 55%	Obfuscation decreases detection rate, and creating metamorphic malware is difficult
Murad et al. [9]	PE	N/A	Custom	N/A	Obfuscation techniques can easily alter malware signature
Holm et al. [10]	PE	Meterpreter backdoor and benign payload	16 different obfuscation tools were tested	54%	Evading antimalware scans is simple

Typically, the selected studies follow the same flow, starting with a set of malware and applying a list of obfuscations separately or in different combinations. For the studies involving Android malware, the malware must first be decompiled before applying obfuscation; usually, this is completed by Apktool. After obfuscations have been used, the malware is passed through a set of antivirus programs. The obfuscation type and the individual antivirus software detection rates often organize the results in these studies. Overall, the average detection rate of obfuscated samples appears to lie somewhere between 55% and 65%. Figure 1 shows the four studies providing final cumulative results.

Nawaz et al. report the highest detection rates by far, with some obfuscation techniques such as repacking and reassembling yielding an average detection rate across antiviruses of around 90 percent. This study reports average detection rates closer to 65 percent when more than one obfuscation technique is

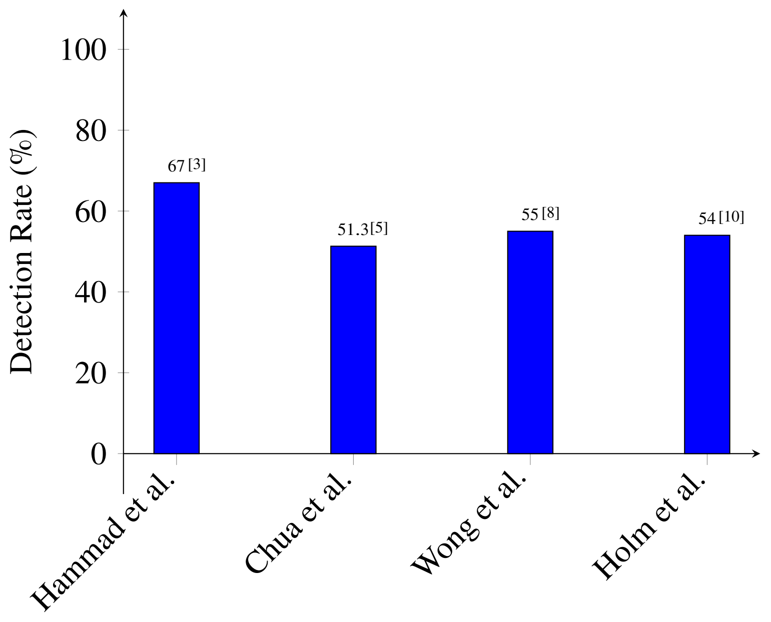

Fig. 1. Signature-based Performance

applied to the same sample. Interestingly, the most effective combinations were the same repacking and reassembling techniques accompanied by encryption, averaging a 39.5% detection rate [4].

B. Neural Network Detection Results

Moving to the neural network-focused studies, we see a much more condensed set of results, although the datasets and experiment setups are still reasonably scattered. Despite using different types of neural networks, three types of input parameters appear the most throughout the selected literature. The first of these three are image-based inputs. These studies convert each malware sample into a grayscale or RGB image based on either binary data as with D. K. A. et al. [11] and Shukla et al. [20] or on system call traces as with Mercaldo et al. [12]. The other two neural networks are those trained on the malware samples' opcodes as with P. Xu et al. [13] and K. Xu et al. [16] and those trained on API calls as with Hou et al. [14], Millar et al. [17], and Namani et al. [19].

Figure 2 organizes the research displayed in Table II by feature type. These features can be further categorized as either statically or dynamically collected.

Statically collected features can be obtained from the sample by examining the binary. Within Table II, the statically collected features in the literature set are found in the studies that use images, opcodes/bytecodes, and API calls. The only exception may be in Namani et al., which obtains the API calls through symbolic execution and could fall somewhere between static and dynamic analysis [19].

The rest of the features (syscall traces, network data, and memory dumps) fall under dynamic analysis, as these features are obtained by executing the potential malware sample in question. Both studies which train neural networks on memory dump data use the same PE malware dataset, CIC-MalMem-2022[1], which contains pre-collected memory dumps rather than whole malware samples [18] [21].

C. Analysis of Research Question

As expected, neural networks perform much better overall than signature-based detection methods in detecting obfuscated malware, with an estimated increase from around 60 percent to as much as 99 percent. As mentioned, the signature-based detection rates cover a relatively wide range of values. Due to results within each study often being organized by the antivirus software used, the results are even more scattered in the literature. A few of these studies did not offer a final average rate but presented results

[1]Dataset available here.

TABLE II
Neural Network Detection

Study	Malware Target	Dataset Source	Obfuscation Tools	NN and training feature type	Detection Rate
D. K. A. et al. [11]	Android	Drebin Project, CIC-InvesAndMal2019	Obfusapk	CNN trained on Markov Images generated from hex code	99%
Mercaldo et al. [12]	Android	Drebin Project	Custom?	NN trained on images generated from system call traces	73%
P. Xu et al. [13]	Android	Drebin, AMD, PRAGuard, AndroZoo	PRAGuard dataset already obfuscated	GNN trained on vector generated from opcodes	99.5%
Hou et al. [14]	Android	Comodo Cloud Security Center	N/A, dynamically collected training features	DBN and SAE trained on API calls	DBN: 96%, SAE: 95%
Busch et al. [15]	Android	CICAndMal2017	N/A	GNN trained on network data	99%
K. Xu et al. [16]	Android	VirusShare, MassVet	DroidChameleon	LSTM trained on bytecode	99.9%
Millar et al. [17]	Android	Drebin	DexProtector	DAN trained on opcodes, API calls, and permissions from dataset	94.8% against all obfuscations, 97% average on whole test set
Mezina et al. [18]	PE	CIC-MalMem-2022	Dataset contains obfuscated samples	CNN trained on dataset's memory dumps	99%
Namani et al. [19]	PE	Anderson and Roth dataset, VirusShare, Malshare, others	N/A	ANN trained on API call sequences	94.6%
Shukla et al. [20]	PE	VirusTotal	Custom?	RNN trained on images generated from samples	90%
Khan [21]	PE	CIC-MalMem-2022	Dataset contains obfuscated samples	ANN trained on dataset's memory dumps	99.7%
Javaheri et al. [22]	PE	Adminus, VirusShare, VirusSign, others	Dataset already contains obfuscated samples	DNN trained on feature set created by genetic algorithm, dynamically collected features such as API calls	Between 91% and 96% depending on dataset - average percentage of 92.8%

organized by antivirus or obfuscation methods. Examples of this include Nawaz et al. [4], Canfora et al. [6], and Christodorescu et al. [7]. No matter the presentation of results, all of the collected studies agree that signature-based detection performs poorly against obfuscated malware.

Neural network detection rates, however, are very high, with one outlying exception in Mercaldo et al. [12]. As shown in Figure 2, the models with opcodes and memory dumps as training features performed the best. Conversely, models trained on API calls are still reliable but noticeably weaker. The outlier, which scored 73 percent, is a CNN trained on images generated on the Android system called traces. This study is included in its category since these images are generated from opcodes or n-grams, not dynamically

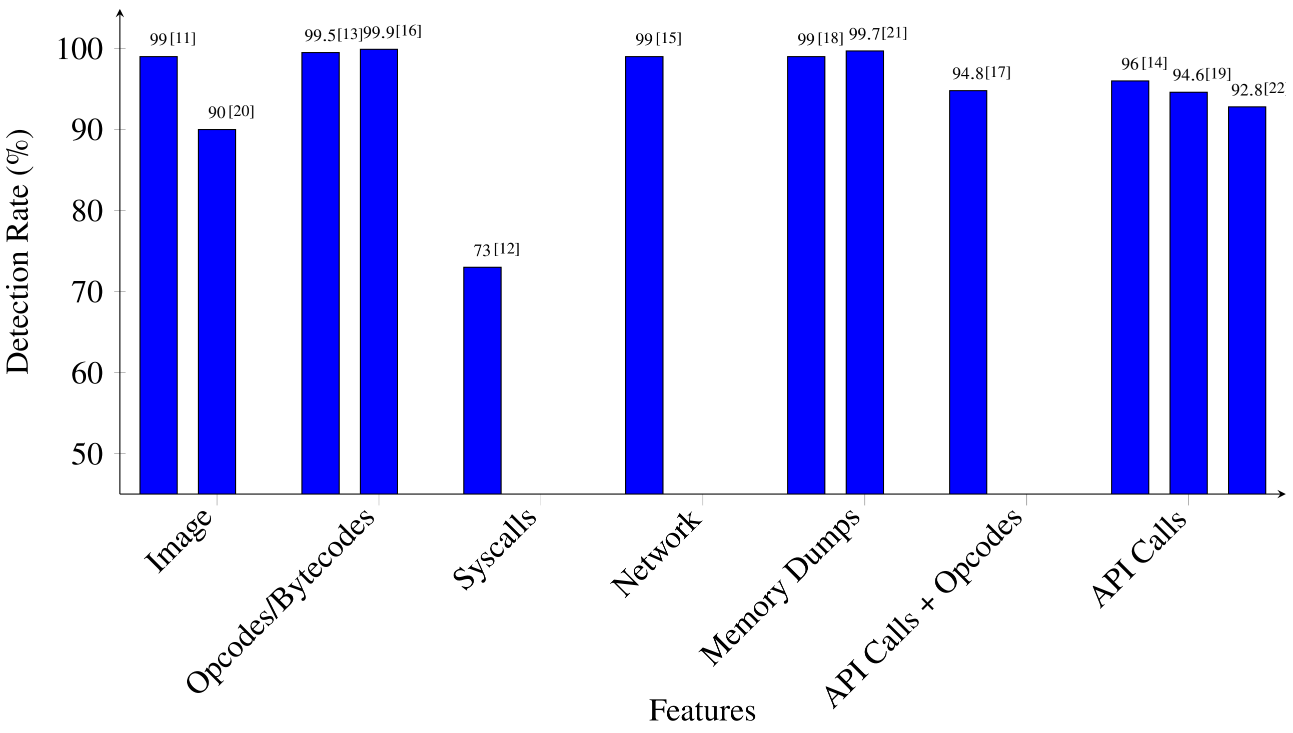

Fig. 2. NN Performance by Training Feature Type

collected features.

A few key characteristics did not appear to affect the outcome of the studies. First, the malware type in question (Android vs. PE) does not have much effect on detection rates, which is valid for signature-based and neural network detection. However, Table I and Table II are sorted by malware type. The other characteristic that does not seem to affect the detection rate is the type of neural network used. ANNs, RNNs, DNNs, and LSTMs are variations of artificial neural networks, but no obvious connection can be found in their results. One definite limitation of this survey is that there is little overlap between the neural networks and training feature types. There is not a single study that uses the same combination of NN and training features.

Regardless, the results of this survey suggest that neural networks contain great potential as an addition to antivirus software. Especially given the exceptional performance of the models trained on statically collected features such as opcodes, it is easy to envision a model integrated into endpoint protection tools. In this context, the scalability of neural networks provides some inherent advantages to the traditional solution of maintaining a signature database.

IV. Conclusion

This survey has collected a variety of literature addressing both signature-based detection and neural network detection of obfuscated malware. As expected, neural network models perform far more reliably than signature-based methods. Recent studies have shown that models trained on malware opcodes or memory dumps are up-and-coming, with 99 percent or better detection rates.

After reviewing the available literature on this subject, it was noticed that a couple of areas need further research. First, research on neural networks trained on obfuscated PE malware needs much improvement. There has been plenty of research on unobfuscated malware detection via deep learning, but the added challenge of obtaining this more advanced malware has deterred work in this area. Secondly, more research is needed on neural networks involving multiple training features. This survey includes one example using API calls and opcodes in Millar et al. [17]. Still, no examples combined statically and dynamically collected features were found.

Overall, this survey has clarified that signature-based detection is not an adequate solution to malware authors' more advanced techniques. In recent studies, neural networks show great potential to solve this issue and perform substantially better than signature-based methods.

References

[1] W. Elsersy, A. Feizollah, and N. Anuar, "The rise of obfuscated android malware and impacts on detection methods," *PeerJ Computer Science*, vol. 8, Mar 2022.

[2] 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.

[3] M. Hammad, J. Garcia, and S. Malek, "A large-scale empirical study on the effects of code obfuscations on android apps and anti-malware products," in *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, (New York, NY, USA), p. 421–431, Association for Computing Machinery, 2018.

[4] U. Nawaz *et al.*, "On the evaluation of android malware detectors against code-obfuscation techniques," *PeerJ Computer Science*, vol. 8, p. e1002, 2022.

[5] M. Chua and V. Balachandran, "Effectiveness of android obfuscation on evading anti-malware," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY '18, (New York, NY, USA), p. 143–145, Association for Computing Machinery, 2018.

[6] G. Canfora, A. Di Sorbo, F. Mercaldo, and C. A. Visaggio, "Obfuscation techniques against signature-based detection: A case study," in *2015 Mobile Systems Technologies Workshop (MST)*, pp. 21–26, 2015.

[7] M. Christodorescu and S. Jha, "Testing malware detectors," in *Proceedings of the 2004 ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA '04, (New York, NY, USA), p. 34–44, Association for Computing Machinery, 2004.

[8] W. Wong and M. Stamp, "Hunting for metamorphic engines," *Journal of Computer Virology*, vol. 2, pp. 211–229, 2006.

[9] K. Murad, S. N.-u.-H. Shirazi, Y. B. Zikria, and N. Ikram, "Evading virus detection using code obfuscation," in *Future Generation Information Technology: Second International Conference, FGIT 2010, Jeju Island, Korea, December 13-15, 2010. Proceedings 2*, pp. 394–401, Springer, 2010.

[10] H. Holm and E. Hyllienmark, "Hide my payload: An empirical study of antimalware evasion tools," in *2023 IEEE International Conference on Big Data (BigData)*, pp. 2989–2998, 2023.

[11] D. K. A., V. P., S. Y. Yerima, A. Bashar, A. David, A. T., A. Antony, A. K. Shavanas, and G. K. T., "Obfuscated malware detection in iot android applications using markov images and cnn," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2756–2766, 2023.

[12] F. Mercaldo, G. Ciaramella, A. Santone, and F. Martinelli, "Obfuscated mobile malware detection by means of dynamic analysis and explainable deep learning," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[13] P. Xu, C. Eckert, and A. Zarras, "Detecting and categorizing android malware with graph neural networks," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, (New York, NY, USA), p. 409–412, Association for Computing Machinery, 2021.

[14] S. Hou, A. Saas, L. Chen, Y. Ye, and T. Bourlai, "Deep neural networks for automatic android malware detection," in *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 803–810, 2017.

[15] J. Busch, A. Kocheturov, V. Tresp, and T. Seidl, "Nf-gnn: Network flow graph neural networks for malware detection and classification," in *Proceedings of the 33rd International Conference on Scientific and Statistical Database Management*, SSDBM '21, (New York, NY, USA), p. 121–132, Association for Computing Machinery, 2021.

[16] K. Xu, Y. Li, R. H. Deng, and K. Chen, "Deeprefiner: Multi-layer android malware detection system applying deep neural networks," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 473–487, 2018.

[17] S. Millar, N. McLaughlin, J. Martinez del Rincon, P. Miller, and Z. Zhao, "Dandroid: A multi-view discriminative adversarial network for obfuscated android malware detection," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, CODASPY '20, (New York, NY, USA), p. 353–364, Association for Computing Machinery, 2020.

[18] A. Mezina and R. Burget, "Obfuscated malware detection using dilated convolutional network," in *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 110–115, 2022.

[19] N. Namani and A. Khan, "Symbolic execution based feature extraction for detection of malware," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–6, 2020.

[20] S. Shukla, G. Kolhe, S. M. P D, and S. Rafatirad, "Stealthy malware detection using rnn-based automated localized feature extraction and classifier," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 590–597, 2019.

[21] L. P. Khan, "Obfuscated malware detection using artificial neural network (ann)," in *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–5, 2023.

[22] D. Javaheri, P. Lalbakhsh, and M. Hosseinzadeh, "A novel method for detecting future generations of targeted and metamorphic malware based on genetic algorithm," *IEEE Access*, vol. 9, pp. 69951–69970, 2021.