A review of: Optimal Feature Configuration for Dynamic Malware Detection

David Escudero García
Research Institute of Applied science in Cybersecurity
Campus de Vegazana s/n, 24071, León, Spain
descg@unileon.es

Noemí DeCastro-García
Universidad de León
Campus de Vegazana s/n, 24071, León, Spain
ncasg@unileon.es

***Abstract*—Applying machine learning techniques to malware detection is a common approach to try to overcome the limitations of signature-based methods. However, it is difficult to engineer a set of features that characterizes the samples properly, especially when various file types may be a vector of infection. In this work, we configure several feature sets for dynamic malware detection extracted from API calls, network activity, signatures from the Cuckoo sandbox report, and some interactions with the file system and registry. We test combinations of these feature sets to ascertain whether they are good enough to distinguish between benign and malicious samples from a dataset containing several file types, obtained from public sources. The datasets present class imbalance to evaluate the model performance on more realistic scenarios in which not many malware samples are available.**

***Index Terms*—Machine Learning, Malware detection, Feature engineering**

Contribution Type: *Published Research*

I. Introduction

In this paper we present a work previously published in the *Computers & Security* journal in 2021 [1].

The threat presented by malware affects both, individual users and organizations. Since signature based approaches require the file to have been previously identified as malware in order to protect the user, research efforts have been directed towards achieving more intelligent detection, usually by applying machine learning.

Most of the research is focused on the detection of malware in executable files from either Windows [2] or Android [3]. However, according to a Verizon report, 45% of malware distribution is carried out through Office documents, with executables at 26% [4]. In addition, datasets with malicious samples usually present class imbalance that may bias the classifiers [5].

Therefore, in this work, we aim to provide an optimal feature configuration for malware detection on different file types (.pdf, .docx, .exe, .html, .xlsx), using machine learning techniques in the presence of class imbalance. The optimal configuration is obtained by analyzing statistically meaningful differences among the results of the models that are constructed with different combinations of feature sets.

We employ dynamic analysis for the construction of the feature sets. We extract different feature sets derived from API calls, network traffic, and the signatures provided by Cuckoo, combined with interaction with the file system and registry, which are commonly used in the literature. We decide to focus on dynamic analysis since, when successful, it provides a more accurate, file-agnostic characterization of sample behavior. This fact allows us to uniformly extract features from several file formats so that a greater number of related problems may be tackled. Otherwise, a different set of static features should be engineered for each file type.

II. Datasets

We have collected 19994 file samples, both benign and malicious, from different public sources. A total of 9999 malicious samples were downloaded from VirusShare [6] and 9995 benign samples were obtained from Digital Corpora [7] and files extracted from local computers. The distribution of file types is described in Table I.

Table I
Distribution of File Types in our File Set

	Word	Excel	HTML	PDF	Executable
Benign	2999	1999	2000	2499	498
Malicious	2769	233	3492	1006	2499

The analysis of files is carried out using the Cuckoo sandbox v2.0.7, using a Windows 7 VM for analysis. The machine was made more vulnerable by disabling several security options such as the firewall and user account control. The JSON report is used to extract API and signatures features and the pcap trace is used for the extraction of network features.

The feature sets used are the following[1]:

- API. Frequencies of categories of calls (file, network, etc.) and of its 2-grams. 342 features in total.
- Network. Summary statistics of the network traffic such as mean and standard deviation of certain quantities such as the bytes sent and received per second and duration of flows. 199 features in total.
- Signatures. Boolean features corresponding to the signatures provided by Cuckoo. In addition, some statistics regarding the interaction with the file system and registry such as the number of files written or types of keys created. 594 features in total.

We have carried out experiments with different combinations of the feature sets to determine which is the most favorable. In addition, we also introduce different degrees of imbalance by modifying the proportion of malware in the dataset. We use proportions of malware of 30%, 20%, 10% and 5%. The combinations of feature sets used are shown in Table II.

[1]Available at https://drive.google.com/drive/folders/173SO6RmkdmWa-5fM7xOz5ZL20eniPMQ-?usp=sharing.

Table II
DESCRIPTION OF FEATURE SETS USED IN EXPERIMENTS

Code	Feature set
$\mathcal{F}_4$	Network + signatures
$\mathcal{F}_5$	API + network + signatures
$\mathcal{F}_6$	API + network
$\mathcal{F}_7$	API + signatures
$\mathcal{F}_i^j$	unbalanced set of features $\mathcal{F}_i$ with a $j\%$ proportion of malicious samples

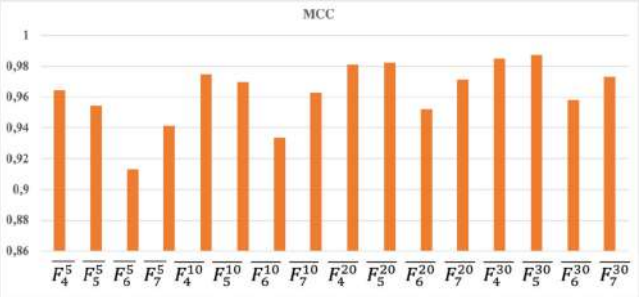

Figure 1. Median MCC for each dataset

III. EXPERIMENTS

We have used the Auto-sklearn library [8] in its 0.7.1 version to construct the models. Auto-sklearn requires that the user sets a time limit for the search of configurations. The maximum time for model construction is set to 15 minutes. We use 10-fold cross-validation as evaluation strategy and use the Matthews correlation coefficient (MCC) as metric, since it is more informative than accuracy in unbalanced datasets. It is defined in Eq. 1.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

This process is repeated 50 times for each dataset. We apply Friedman's test followed by tests for two-paired samples to determine whether there are significant differences between the performances of different combinations of feature sets.

IV. RESULTS

The results of the statistical tests show that there are statistically significant differences between the MCC achieved for different feature sets, so we proceed to compare their prediction performance.

The MCC achieved in each dataset is shown in Fig. 1. Usually, we can consider a model good enough if it achieves a MCC greater than 0.95, without taking into account specific application requirements regarding false positives and false negatives. In our experiments, datasets $\mathcal{F}_4$ and $\mathcal{F}_5$ exceed that threshold for all degrees of imbalance and are in general the best performant. $\mathcal{F}_4$ achieves better results than $\mathcal{F}_5$ in greater degrees of imbalance despite both sets sharing network and signatures features, which signals that the API set may be redundant with others given its low-level nature.

In order to give a more accurate vision of model performance, Fig. 2 shows the true positive rate (sensitivity) and true negative rate (specificity) achieved for each dataset. A model

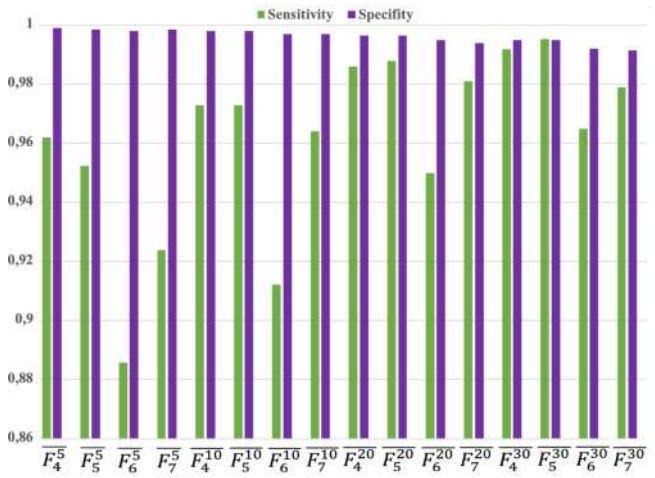

Figure 2. Median sensitivity and specificity for each dataset

with high sensitivity will detect most malicious samples and will not have many false negatives. A high-specificity model will correctly categorize benign samples and will not generate many false positives. As can be seen, the obtained models achieve higher specificity, nearing 1, which could be expected since benign samples are the majority class. In addition, sensitivity is also over 0.95 even at the highest degree of imbalance with $\mathcal{F}_4$ and $\mathcal{F}_5$.

V. CONCLUSIONS

In the original work we extracted several feature sets from dynamic analysis and tested their effectiveness in the problem of detecting malware across several file types in the present of class imbalance. The results show that the selected features are effective at detecting malware in our setting. In particular, the combination of network and signatures set achieves the highest results, with an MCC higher than 0.95 for all degrees of imbalance.

ACKNOWLEDGEMENTS

This work was partially supported by the Spanish National Cybersecurity Institute (INCIBE) under contract Art.83, key: X54. Also, we thank Ángel Luis Muñoz Castañeda for his advice regarding the manuscript.

REFERENCES

[1] D. Escudero-García and N. deCastro García, "Optimal feature configuration for dynamic malware detection," *Computers & Security*, 2021.
[2] W. Han, J. Xue, Y. Wang, Z. Liu, and Z. Kong, "Malinsight: A systematic profiling based malware detection framework," *Journal of Network and Computer Applications*, vol. 125, pp. 236–250, Jan. 2019.
[3] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning," *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 114–123, Feb. 2016.
[4] Verizon, "2019 data breach investigations report," 2020. [Online]. Available: https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf
[5] 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, Mar. 2020.
[6] Corvus Forensics, "Virusshare," 2011, accessed: Dec. 2019. [Online]. Available: https://virusshare.com/
[7] Digital Corpora, "Govdocs1 — (nearly) 1 million freely-redistributable files," 2018. [Online]. Available: https://digitalcorpora.org/corpora/files
[8] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. of 28 Conf. in Adv. in Neural Inf. Process. Syst.*, 2015, pp. 2962–2970.