

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 78 (2016) 667 – 674

Procedia
Computer Science

International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015,
Nagpur, INDIA

Grouping the executables to detect malwares with high accuracy

Sanjay K. Sahay^a, Ashu Sharma^{b,*}

^aAssistant Professor, Dept of Computer Science and Information System, BITS PILANI, K. K. Birla Goa Campus, India

^bResearch Scholar, Dept of Computer Science and Information System, BITS PILANI, K. K. Birla Goa Campus, India

Abstract

The metamorphic malware variants with the same malicious behavior (family), can obfuscate themselves to look different from each other. This variation in structure lead to a huge signature database for traditional signature matching techniques to detect them. In order to effective and efficient detection of malwares in large amounts of executables, we need to partition these files into groups which can identify their respective families. In addition, the grouping criteria should be chosen such a way that, it can also be applied to unknown files encounter on computer for classification. This paper discusses the study of malwares and benign executables in groups to detect unknown malwares with high accuracy. We studied sizes of malwares generated by three popular second generation malwares (metamorphic malwares) creator kits viz. G2, PS-MPC and NGVCK, and observed that the size variation in any two generated malwares from same kit is not much. Hence we grouped the executables on the basis of malware sizes by using Optimal k-Means Clustering algorithm and used these obtained groups to select promising features for training (Random forest, J48, LMT, FT and NBT) classifiers to detect variants of malwares or unknown malwares. We find that detection of malwares on the basis of their respected file sizes gives accuracy up to 99.11% from the classifiers.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of organizing committee of the ICISP2015

Keywords: Anti-Malware; Static Malware Analysis; WEKA; Machine Learning

* Corresponding author. Tel.: +91-8975805861.

E-mail address: p2012011@goa.bits-pilani.ac.in

1. Introduction

As new variants of malwares getting evolve every day, malwares defense becoming increasingly difficult task in detecting malware and protecting computers systems from them¹. Recently 11 zero-day vulnerabilities reported during the month of August while 6 of these were reported in industrial control systems². Even state sponsored highly skilled hackers are developing customized malwares to disrupt industries and for military espionage³. Many of countries continue to incur most costly data breaches. Among them two countries had the highest cost from data breach⁴ (the U.S. at \$5.4 million and Germany at \$4.8 million).

Anti-malware industries are facing a major challenge of continuously increase of huge data, which need to be checked for potential malicious content. Microsoft reports that there real-time detection anti-malware products are present on over 160 Million computing devices throughout the globe and they daily analyze tens of millions of data files as potential malware⁵. Reason behind these high volumes of different files is that the malware authors introduce metamorphism to the malicious components. Metamorphic malware represent the next class of virus that can create an entirely new variant after reproduction³. The new variant produced is in no-way similar to the original variant which lead a huge increase in the malwares count.

In order to detect them with high accuracy, we need to group them to identify their respective families. In addition, such grouping criteria may be applied to new test executables to classify it to malware. In this paper we studied three popular second generation malwares creator kits viz. G2, PS-MPC and NGVCK and found that the size variation in any two generated malwares from same kit does not differ much. Hence in this work we grouped the executables on the basis of malware sizes by using optimal k-Means Clustering algorithm and promising features are selected separately from each groups. Further these obtained features are tested on random forest, J48, Logistic Model Trees (LMT), functional trees (FT) and naive bayes tree (NBT) classifier using machine learning technique.

The paper is organized as follow, in next section related work is discussed, In section 3 we present our approach, The section 4 discuss the experimental results and finally section 5 contains the conclusion and future directions.

2. Related work

The first virus was created in 1970⁶ and since then there is a strong contest between the attackers and defenders. To combat the threats/attacks from the second generation malwares, Schultz et al. (2001) was the first to introduce the concept of data mining to classify the malwares⁷. In 2005 Karimet al.⁸ addressed the tracking of malware evolution based on opcode sequences and permutations. O. Henchiri et al.(2006) proposed a hierarchical feature extraction algorithm and used ID3, j48, Naive Bayes and SMO classifier and obtained maximum of 92.56% accuracy⁹. In year 2005, Karimet al.⁸ addressed the tracking of malware evolution based on opcode sequences and permutations. O. Henchiri et al.(2006) proposed a hierarchical feature extraction algorithm and used ID3, j48, Naive Bayes and SMO classifier and obtained maximum of 92.56% accuracy⁹. Bilar (2007) uses small dataset to examine the opcode frequency distribution difference between malicious and benign code¹⁰ and observed that some opcodes seen to be a stronger predictor of frequency variation. In 2008, Yanfang Ye et. al.¹¹ applied association rules on API execution sequences for classifying the malwares. In 2008, Tian et al.¹² classified the Trojan using function length frequency and shown that the function length along with its frequency is significant in identifying malware family and can be combined with other features for fast and scalable malware classification. Moskovitch et al.¹³ compared the different classifiers by byte-sequence n-grams (3, 4, 5 or 6). Among the classifiers they studied BDT, DT and ANN out-performed NB, BNB and SVM classifiers, exhibiting lower false positive rates. In year 2008, Siddiquiet al.¹⁴ used variable length instruction sequence for detecting worms in the wild. They tested their method on a data set of 2774 (1444 worms and 1330 benign files) and got 95.6% detection accuracy. In 2009 S. Momina Tabish¹⁵ used 13 different stational features computed on 1, 2, 3 and 4 gram by analyzing byte-level file content for classification of malwares. In year 2010, Bilal Mehdi et. al.¹⁶ used hyper grams (generalized n-gram) and obtained 87.85% detection accuracy and claimed no false alarm. ChatchaiLiangboonprakong et al. (2013) proposed a classification of malware families based on N-grams sequential pattern features¹⁷. They used DT, ANN and SVM

classifier and obtained good accuracy. Santos et al. in year 2011 pointed out that supervised learning requires a significant amount of labeled executables for both malware and benign programs, which is difficult to obtain, hence they proposed a semi-supervised learning method for detecting unknown malwares, which does not require a large amount of labeled data¹⁸. They obtained 86% of accuracy by labeling only 50% of the selected data set. In subsequent paper¹⁹ in 2013, they used Term Frequency for modeling different classifiers and found that SVM outperform with accuracy of 92.92% and 95.90% respectively for one opcode and two opcode sequence length respectively. Recently (2014) Zahra Salehi et al. construct feature set by extracting API calls used in the executables for the classification of malwares²⁰.

3. Our approach

Fig. 1 represents the procedure to partition the dataset in groups, finding the promising features from each formed group and the classification of unknown malwares by using Random forest²¹, J48²², LMT²³, FT²⁴ and NBT²⁵ classifiers available in WEKA tool²⁶.

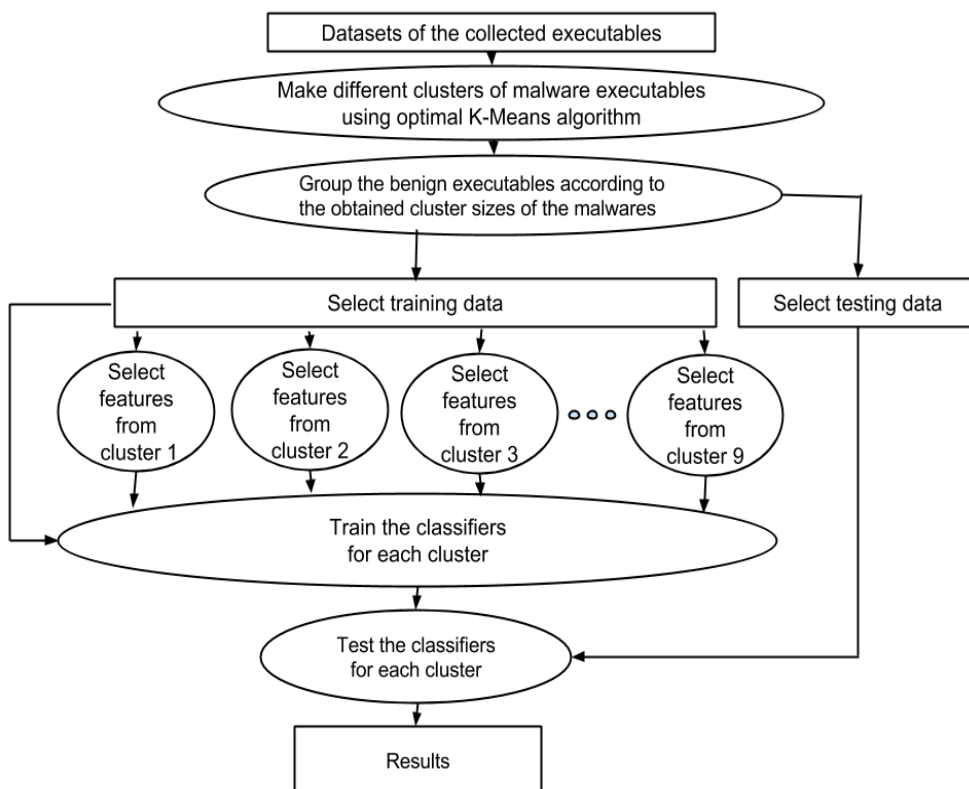


Fig.1.:Flow chart for the proposed method.

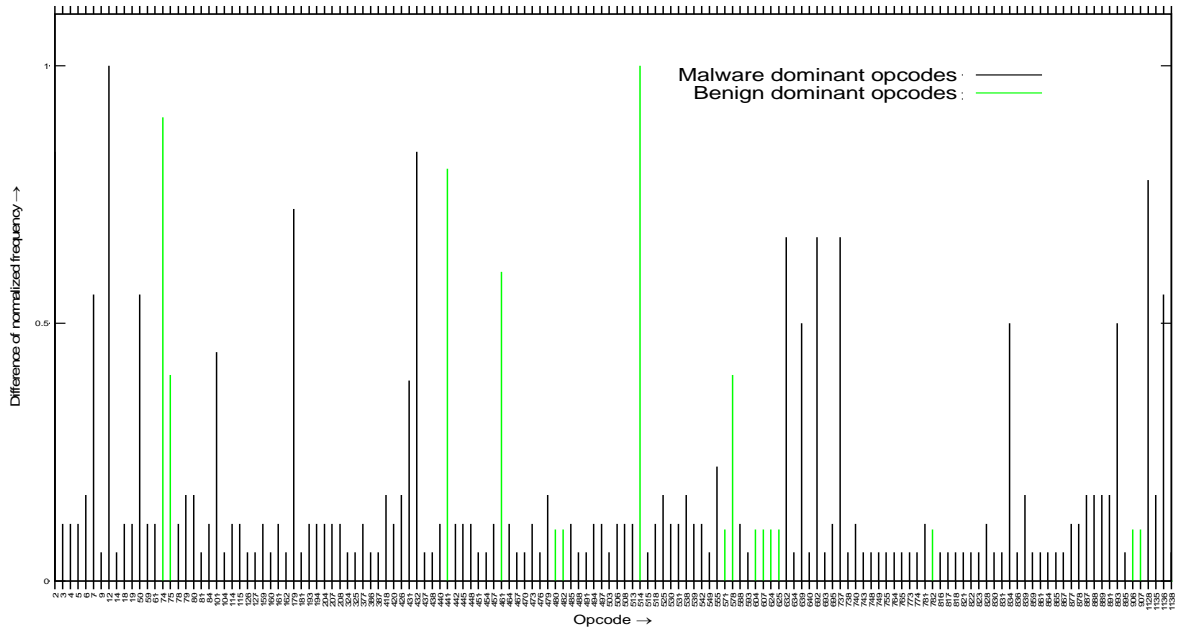


Fig.2.:Opcodes that have high difference of occurrence in malware and benign executables.

3.1. Data preprocessing

For the experimental analysis, we downloaded 11368 malwares from malacia-project²⁷ and collected 3151 benign programs (also verified from virustotal.com²⁸) from different systems.

Algorithm1: Selection of the Promising Features

INPUT: Malwares and benign assembly codes, $N_b \rightarrow$ Total No. of benign in the cluster, $N_m \rightarrow$ Total No. of malware in the cluster, $N \rightarrow$ Required No. of features.

OUTPUT: Features for the analysis.

BEGIN

for all Malwares and benign data **do**

Compute the sum of Normalized frequency(f_i) of each opcode o_j .

$$S_{F_m}(o_j) = \left(\sum f_i(o_j) / \text{Max}(f(o_j)) \right) / N_m$$

$$S_{F_b}(o_j) = \left(\sum f_i(o_j) / \text{Max}(f(o_j)) \right) / N_b$$

end for

for all opcode o_j **do**

Compute the difference $D(o_j)$ between the $S_{F_m}(o_j)$ and $S_{F_b}(o_j)$ for each opcode.

$$D(o_j) = \left| S_{F_m}(o_j) - S_{F_b}(o_j) \right|$$

end for

return N number of opcodes with highest $D(o_j)$.

For the analysis we disassemble all collected executables to their assembly codes by *objdump* utility available in the Linux system and found 1147 unique opcodes, which we labeled with a fixed integer. viz. 1 \leftarrow aaa, 2 \leftarrow aad, ..., 1147 \leftarrow xsha256.

We study the obtained opcodes occurrence in malwares and benign executable and found that many opcodes occurrences in malwares differ significantly from benign program, and vice versa (fig. 2). Hence, we obtained the promising features by computing the difference of normalized opcodes frequency between benign and malware executables as given in the algo. 1.

3.2. Partitioning the executables in dataset

We studied the malwares sizes generated by popular metamorphic malware generator kits viz. NGVCK²⁹, PS-MPC³⁰ and G2³¹ and found that size of the malwares generated by any one of the kits does not vary much (fig. 3). For efficient analysis and classification of advanced malwares in large amounts of executables, we used size of an executable as criteria for grouping the dataset. We partitioned the collected dataset into 9 groups by using optimal k-Means Clustering algorithm. The number of clusters (value of K) is obtained by the Bayesian information criterion³².

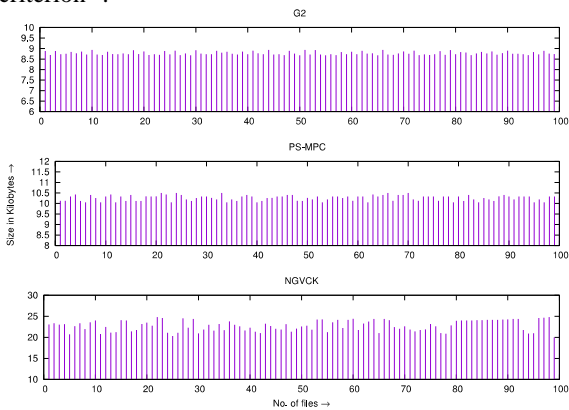


Fig.3.:Variation in the size of malwares generated by three different malware generators.

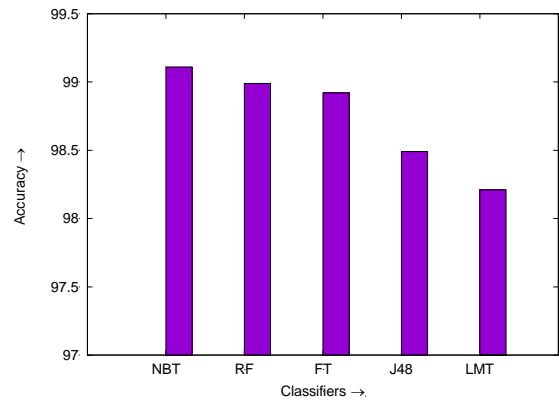


Fig.4.:Detection accuracy by selecting most promising number of features

4. Experimental analysis and results

We have used machine learning to train and test the Random forest, J48, LMT, FT and NBT classifiers available

Table 1. Number of executables in each cluster of malwares and benign dataset.

Cluster	No. of malwares for training	No. of benign for training	No. of malwares for testing	No. of benign for testing
1	322	43	55	18
2	1234	20	221	9
3	1489	20	265	9
4	714	71	128	13
5	335	2227	61	402
6	886	36	158	11
7	2716	40	481	11
8	1148	33	204	11
9	18	156	4	21
Total	8862	2646	1577	505

in WEKA tool (a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functionality²⁶).

We have obtained 9 groups from the datasets, which are further divided into two sets. One of the sets is used to training the classifiers and other one is used to test the detection accuracy of trained classifiers. The training set consists of 8862 malwares and 2646 benign executables and for testing the classifiers 1577 malware 505 benign executables are taken. The number of malwares and benign executables for training and testing the classifiers are given in table 1. In this for robust results, we ensure that at least 15% of the executables in the cluster which are not used for training purpose are taken for the testing of the classifiers.

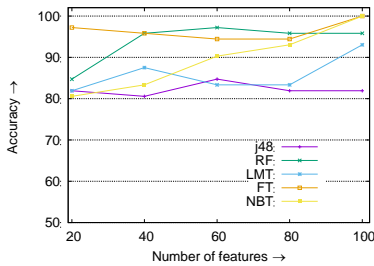


Fig.5.:Detection accuracy obtained by the classifiers on group 1.

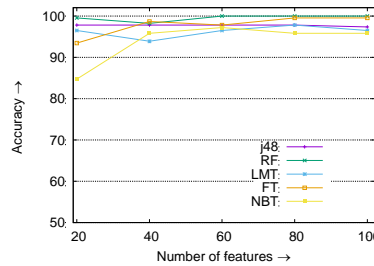


Fig.6.:Detection accuracy obtained by the classifiers on group 2.

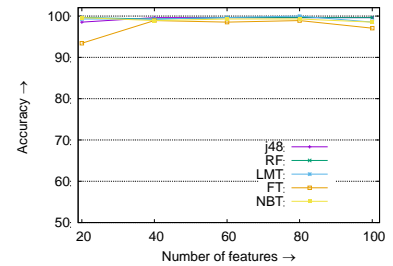


Fig.7: Detection accuracy obtained by the classifiers on group 3.

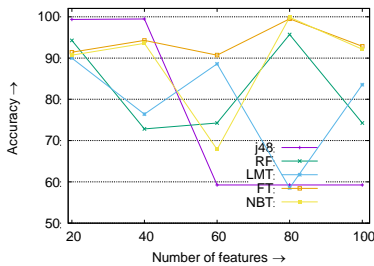


Fig.8: Detection accuracy obtained by the classifiers on group 4.

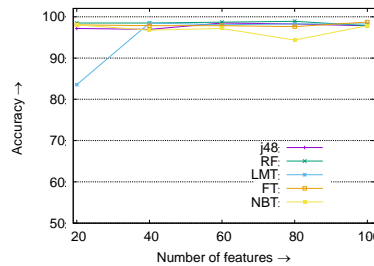


Fig.9: Detection accuracy obtained by the classifiers on group 5.

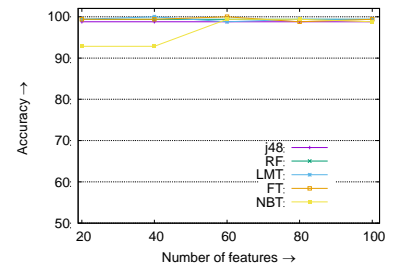


Fig.10: Detection accuracy obtained by the classifiers on group 6.

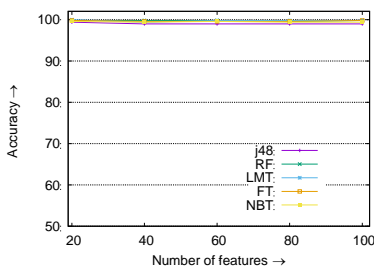


Fig.11: Detection accuracy obtained by the classifiers on group 7.

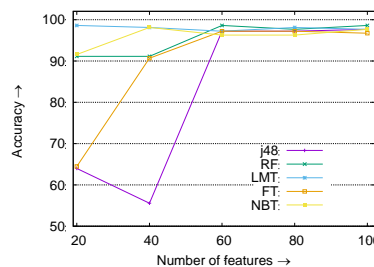


Fig.12: Detection accuracy obtained by the classifiers on group 8.

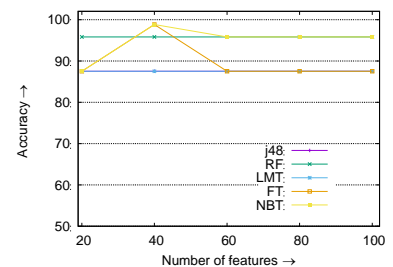


Fig.13: Detection accuracy obtained by the classifiers on group 9.

To train the classifiers, first we have used the feature selection algorithm (1) to find the promising features for the detection of malwares for every partitioned group 1-9 and then trained the classifier. To measure the effectiveness of the five classifiers we used different number of features (20, 40, 60, 100) for the classification and monitored the accuracy of classifiers for each groups with providing the respective testing executables of the group. The detection accuracy of the classifier is calculated by the equation.

$$Accuracy(\%) = \frac{TP + TN}{TM + TB} \times 100 \quad (1)$$

where,

TP→ True positive, the number of malwares correctly classified.

TN→ True negative, the number of benign correctly classified.

TM→ Total number of malwares.

TB→ Total number of benign.

The plots (5 - 13) shows the performance of classifiers for each group with respect to different numbers of features provided to the classifiers. We took the best required features from each cluster and find that all the classifier overall performed with correctness of more than 98.21% of detection accuracy. NBT performed better then the other with 99.11% of detection accuracy (shown in fig. 4).

5. Conclusion

From the literature, we found that the obfuscation techniques in malware generation are becoming crucial for the classical signature matching approaches used in malwares detection systems. In order to be efficient in analyzing and classifying large amounts of executables for detection of unknown or metamorphic malwares, we have studied the malwares generated by three popular malware generator kits. We found the variation of size of malwares generated by these kits does not vary much. Thus on the basis of findings, we partitioning executables in 9 groups using Optimal k-Means Clustering algorithm on the sizes of malwares as a criteria. Then we discussed a statical analysis approach to detect unknown malwares for windows operating system by selecting features for each group separately by feature selection approach based on the difference of opcode occurrence among benign and malwares executable. We used a well known classifiers viz. Random forest, J48, LMT, FT and NBT for the classification of malwares and found that all of them perform with more than 98% of detection where as NBT give highest accuracy of 99.11%. The experiments suggest that the research direction is promising and the possible future work can be in-depth study of malware generator kits by which we can able to keep a family of malware in one partitioned group and the classifier can attain the detection of unknown malwares with more accuracy.

Acknowledgements

Mr. Ashu Sharma is thankful to BITS, Pilani, K.K. Birla Goa Campus for the support to carry out his work through Ph.D scholarship No. Ph603226/Jul. 2012/01. We are also thankful to IUCAA, Pune for providing hospitality and computation facility where part of the work was carried out.

References

1. McAfee labs threats report, Tech. rep., McAfee Labs (May 2015).
2. Internet security threat report, Tech. rep., Symantec Corporation (2015).
3. A. Sharma, S. Sahay, Evolution and detection of polymorphic and metamorphic malwares: A survey, *International Journal of Computer Applications* 90 (2) (2014) 7.
4. S. Corporation, Internet security threat report 2014, Tech. rep., Symantec (April 2014).
5. M. Corporation, Microsoft security intelligence report, Tech. rep., Microsoft (june 2014).
6. P. Szor, *The art of computer virus research and defense*, Pearson Education, 2005.
7. M. G. Schultz, E. Eskin, E. Zadok, S. J. Stolfo, Data mining methods for detection of new malicious executables, in: *Security and Privacy*, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on, IEEE, 2001, pp. 38-49.
8. M. E. Karim, A. Walenstein, A. Lakhota, L. Parida, Malware phylogeny generation using permutations of code, *Journal in Computer Virology* 1 (1-2) (2005) 13-23.

9. O. Henchiri, N. Japkowicz, A feature selection and evaluation scheme for computer virus detection, in: *Data Mining, 2006. ICDM'06. Sixth International Conference on*, IEEE, 2006, pp. 891-895.
10. D. Bilar, Opcodes as predictor for malware, *International Journal of Electronic Security and Digital Forensics* 1 (2) (2007) 156-168.
11. Y. Ye, D. Wang, T. Li, D. Ye, Q. Jiang, An intelligent pe-malware detection system based on association mining, *Journal in computer virology* 4 (4) (2008) 323-334.
12. R. Tian, L. M. Batten, S. Versteeg, Function length as a tool for malware classification, in: *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, IEEE, 2008, pp. 69-76.
13. R. Moskovitch, C. Feher, N. Tzachar, E. Berger, M. Gitelman, S. Dolev, Y. Elovici, Unknown malcode detection using opcode representation, in: *Intelligence and Security Informatics*, Springer, 2008, pp. 204-215.
14. M. Siddiqui, M. C. Wang, J. Lee, Detecting internet worms using data mining techniques, *Journal of Systemics, Cybernetics and Informatics* 6 (6) (2008) 48-53.
15. S. M. Tabish, M. Z. Shafiq, M. Farooq, Malware detection using statistical analysis of byte-level file content, in: *Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics*, ACM, 2009, pp. 23-31.
16. B. Mehdi, F. Ahmed, S. A. Khayyam, M. Farooq, Towards a theory of generalizing system call representation for in-execution malware detection, in: *Communications (ICC), 2010 IEEE International Conference on*, IEEE, 2010, pp. 1-5.
17. C. Liangboonprakong, O. Sornil, Classification of malware families based on n-grams sequential pattern features, in: *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, IEEE, 2013, pp. 777-782.
18. I. Santos, J. Nieves, P. G. Bringas, Semi-supervised learning for unknown malware detection, in: *International Symposium on Distributed Computing and Artificial Intelligence*, Springer, 2011, pp. 415-422.
19. I. Santos, F. Brezo, X. Ugarte-Pedrero, P. G. Bringas, Opcode sequences as representation of executables for data-mining-based unknown malware detection, *Information Sciences* 231 (2013) 64-82.
20. Z. Salehi, A. Sami, M. Ghiasi, Using feature generation from api calls for malware detection, *Computer Fraud & Security* 2014 (9) (2014) 9-18.
21. L. Breiman, Random forests, *Machine learning* 45 (1) (2001) 5-32.
22. N. Bhargava, G. Sharma, R. Bhargava, M. Mathuria, Decision tree analysis on j48 algorithm for data mining, *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering* 3 (6).
23. N. Landwehr, M. Hall, E. Frank, Logistic model trees, *Machine Learning* 59 (1-2) (2005) 161-205.
24. N. Landwehr, M. Hall, E. Frank, Logistic model trees, in: *Machine Learning: ECML 2003*, Springer, 2003, pp. 241-252.
25. R. Kohavi, Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid., in: *KDD, Citeseer, 1996*, pp. 202-207.
26. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, *ACM SIGKDD explorations newsletter* 11 (1) (2009) 10-18.
27. A. Nappa, M. Z. Rafique, J. Caballero, Driving in the cloud: An analysis of drive-by download operations and abuse reporting, in: *Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, 2013, pp. 1-20.
28. J. Canto, M. Dacier, E. Kirda, C. Leita, Large scale malware collection: lessons learned, in: *IEEE SRDS Workshop on Sharing Field Data and Experiment Measurements on Resilience of Distributed Computing Systems, 2008*.
29. A. Venkatesan, Code obfuscation and virus detection, Ph.D. thesis, San Jose State University (2008).
30. P. Beaucamps, Advanced metamorphic techniques in computer viruses, in: *International Conference on Computer, Electrical, and Systems Science, and Engineering-CESSE'07, 2007*.
31. T. H. Austin, E. Filiol, S. Josse, M. Stamp, Exploring hidden markov models for virus analysis: a semantic approach, in: *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, IEEE, 2013, pp. 5039-5048.
32. S. S. Chen, P. S. Gopalakrishnan, Clustering via the bayesian information criterion with applications in speech recognition, in: *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, Vol. 2, IEEE, 1998, pp. 645-648.