METAMORPHIC MALWARE DETECTION USING MACHINE LEARNING

MOHAMMED HASAN ALI AHMED ALI

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master's Project Report (By course work) (Electronics & Communications)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2020

DEDICATION

This project report is dedicated to:

The Almighty Allah, for bestowing me the guidance and blessings. To my beloved mother Jamilah and my father Hasan for their unconditional

love and unlimited support,

My brother and sisters, Ali, Bayan, Afnan, Halemah, Alla, and Radeeah.

ACKNOWLEDGEMENT

First and foremost, all gratitude to the omnipresent Allah for giving me the strength through my prayers and to plod me on despite the difficult situations I passed through till my graduation. I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Ismahani Ismail for her support, encouragement, assistance and guidance.

ABSTRACT

Commercially available antivirus software relies on a traditional malware detection technique known as signature-based malware detection which fails to counter unknown signatures of malicious software. Obfuscated malware such as polymorphic or metamorphic are capable of generating a unique signature at each time of executing the malware code to avoid being detected by antivirus software. However, some imperative portions of the malicious code remain unaltered after the obfuscation process. This research project proposes an alternative method of malware detection by utilizing machine learning techniques in which informative textual string attributes were employed as features in with the aim to increase the classifier accuracy and to decrease the computational overhead. In order to develop machine learning classifier models, two phases of learning were applied which are training and testing phases. In this project, benign and malware executable files were collected, then converted to assembly code using disassembler such as IDA Pro disassembler, and then preprocessed to determine the most significant features to aid the machine learning training stage. In addition, part of the collected dataset was obfuscated to be used as testing files in order to test the accuracy of the classifier. The obtained results generated by WEKA platform show that the generative classifier model based on the SMO algorithm has the highest accuracy level and the lowest time taken to build the model. Exploiting the most important textual strings as machine learning training features reduced the computational complexity in terms of the time taken to generate the model and the computing resources such as processing power and memory space. Malware classification using machine learning algorithms proofed to be more effective than traditional signature-based antivirus scanners.

ABSTRAK

Perisian antivirus yang tersedia secara komersial bergantung pada teknik pengesanan malware tradisional dikenali sebagai pengesanan malware berdasarkan tandatangan yang gagal mengatasi tanda tangan yang tidak diketahui perisian berbahaya. Malware yang disamarkan seperti polimorfik atau metamorfik mampu menghasilkan tandatangan unik pada setiap masa melaksanakan kod perisian hasad untuk mengelakkan daripada dikesan oleh perisian antivirus. Walau bagaimanapun, beberapa bahagian penting dari kod jahat tetap tidak berubah selepas proses pengaburan. Projek penyelidikan ini mencadangkan kaedah alternatif malware pengesanan dengan menggunakan teknik pembelajaran mesin di mana atribut rentetan teks bermaklumat digunakan sebagai ciri dengan tujuan untuk meningkatkan ketepatan pengelasan dan mengurangkan overhed pengiraan. Untuk mengembangkan model pengkelasan pembelajaran mesin, dua fasa pembelajaran diterapkan iaitu fasa latihan dan ujian. Dalam projek ini, fail tidak berbahaya dan fail perisian hasad yang dapat dikumpulkan, kemudian ditukarkan ke kod pemasangan dengan menggunakan pembongkar seperti IDA Pro disassembler, dan kemudian memproses untuk menentukan ciri yang paling penting untuk membantu tahap latihan pembelajaran mesin. Selain itu, sebahagian dari kumpulan data yang dikumpulkan dikaburkan untuk digunakan sebagai menguji fail untuk menguji ketepatan pengelasan. Hasil yang diperoleh dihasilkan oleh Platform WEKA menunjukkan bahawa model pengkelasan generatif berdasarkan algoritma SMO mempunyai tahap ketepatan tertinggi dan masa terendah yang diambil untuk membina model. Mengeksploitasi rentetan teks sebagai ciri latihan pembelajaran mesin mengurangkan kerumitan komputasi dari segi masa yang diambil untuk menghasilkan model dan sumber pengkomputeran seperti daya pemprosesan dan ruang memori. Klasifikasi perisian hasad yang menggunakan algoritma pembelajaran mesin terbukti lebih berkesan daripada pengimbas antivirus berasaskan tandatangan tradisional.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GUI	-	Graphical User Interface
IDA	-	Interactive Disassembler
TPR	-	True Positive Rate
TNR	-	True Negative Rate
FNR	-	False Negative Rate
FPR	-	False Positive Rate
SMO	-	Sequential minimal optimization
NGVCK	-	Next Generation Virus Creation Kit
WEKA	-	Waikato Environment for Knowledge Analysis

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The usage of the internet has turned out to be an integral component of our life as the number of people using various services that are provided on the internet is increasing. The internet as we know it today has progressed from a simple communication structure to a group of interrelated data sources which permit, alternative ways of social communications and various retailers to promote and sell their products and services. Services and products advertising or electronic commerce are illustrations of the internet commercial features. In the virtual domain, there are individuals with malicious intentions, who attempt to gain wealth by attacking sincere users and taking advantage of them on every occasion where money is concerned. Network and computer security are major interests for the processes of organizations such as corporations, banks, governments, as well as individuals, in which immense quantity of crucial data are constantly extracted. Safeguarding these important data contained in storage devices of computers or while data transactions between different parties has become an essential duty [1].

Therefore, efficient and precise detection mechanisms have become a necessity to identify malware attacks targeting both computer and network systems. Formerly, worm named Slammer targeted several computers across the globe in a brief instance of time in which it caused massive economic damage [2].

Attacks initiated by malware have become more damaging to the data of both individuals and institutions which resulted in significant losses. For instance, in 2009 a malware known as Zeus attacked more than 3.5 million computer systems of banks in the United States of America. The estimated financial losses of the 2009 incident was approximately about 3 billion US dollars. New malwares are generated in high

rate which causing a challenge for traditional malware detection methods. A statistical analysis conducted by Mcafee showed that more than 80 million malwares were captured in 2013, and in 2014 the number has increased to 140 million malwares [3]. In 2017, a statistical study performed by GDATA program revealed that in every 4 seconds, new malware variants are generated. In 2018, AV-Test which is a well-known institution that examine the performance of anti-malware software by various vendors, revealed that about 20 million samples of new malware were captured in the first three months of the year of 2018 [4]. Figure 1.1 depicts the AV-Test statistical analysis of the number of malware samples for the past ten years. Actually, the chief problem of detecting malwares is not only the huge number of malwares collected but also the sophisticated structure of the malware where the malware creators use obfuscation techniques to cover the main functionality of the malware for the purpose of evading to be detected by anti-virus software [5].

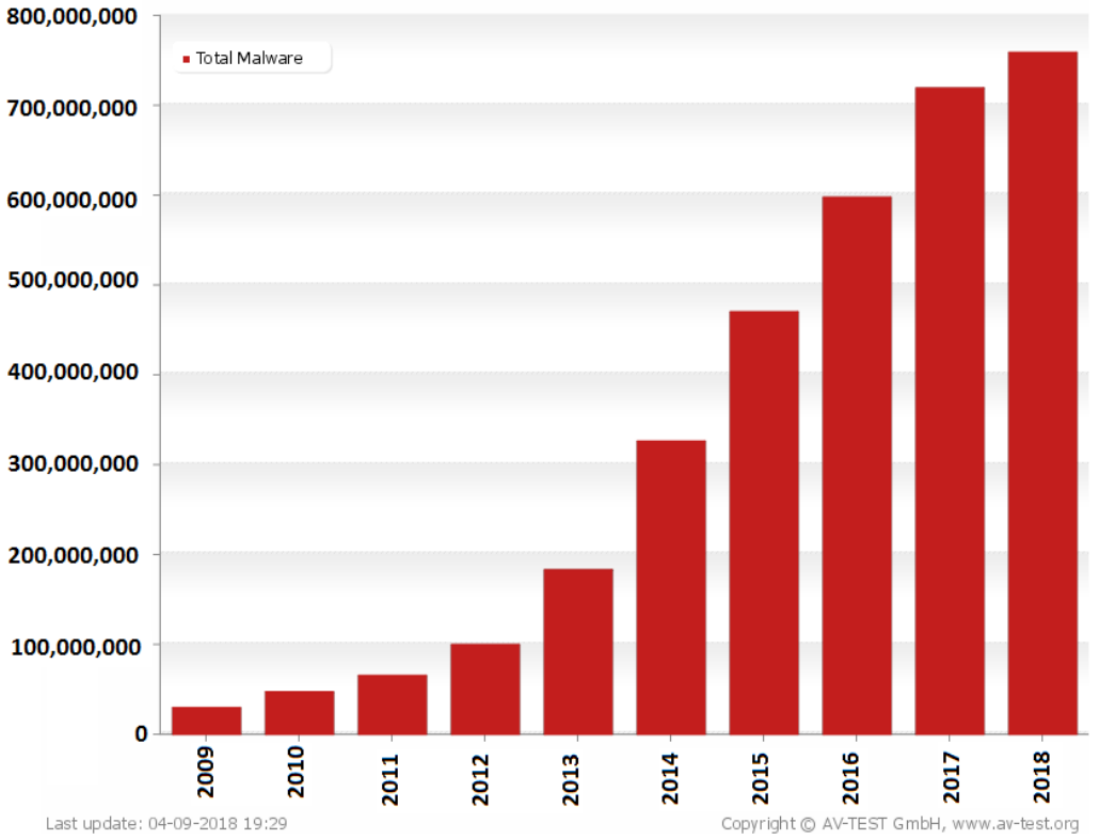

Figure 1.1: Malware statistics for the past ten yeas which carried out by AV-Test

Hackers make advantage of weaknesses in operating system architectures, web browsers, and online services, or exploit social engineering methods to motivate individuals to execute malicious software for the purpose of spreading them. Hackers utilize obfuscation methods such as reassigning registers, insertion of inexecutable code, reordering of the subroutine, transposition of code, integration of code, and substitution of instructions to avoid to be detected by conventional lines of defense such as gateways, firewalls, anti-virus software which relies on signature-based methods [6]. This method demands the anti-virus vendors to supply a record of signatures of already analyzed malware samples to be then compared to possible attacks. Thus, commercially available anti-virus programs are uncapable to detect formerly unobserved malevolent executables.

Since obfuscated malware executables alter continuously their signatures, and hence traditional techniques are not capable to detect their presence. Machine learning techniques rely on learning from previous experiences in order to predict and classify future events. These methods might not substitute the conventional techniques of detections, but they can aid them [10]. In general, machine learning methods demand more computational resources compared to the traditional methods of detections. Additionally, one of the limitations of machine learning is the occurrence of false alarms. Also, ineffective selection of features and redundant features used to build classifier models are additional downsides of machine learning techniques [11]. Therefore, the aim of this work is to use only significant features for generating machine learning classifier models to reduce the feature space and the computational overhead to improve the classification accuracy.

1.2 Problem statement

Current anti-malware software provided by various vendors utilizes signature-based detection scheme. Nonetheless, malicious code is constantly evolving and hackers are exploiting advanced obfuscation techniques such as polymorphism and metamorphism to evade detection by signature based anti-malware software [12]. Self-mutating malwares such as metamorphic malware is automatically changing its signature whenever the malware is executed. This kind of obfuscated malware makes

it difficult for traditional techniques which rely on signature matching to detect it [13]. After the obfuscation operation is initiated, a successor metamorphic malware is produced which carries inherited sections of code from its predecessor. Hackers have a practice to recycle previously known sections of code, as a result of that practice some characteristics from the predecessor metamorphic malware will remain unaltered in new self-mutated malware. It is believed that a complete metamorphic malware transmutation is not possible because complete transmutation will change the function of successor metamorphic malware which is not the goal of the hackers [14]. Therefore, different malware variants originated from a single malware will have similar composition of unaltered sections of code.

As self-mutated malware has the ability to avoid to be detected be conventional methods of detection, hence, machine learning algorithms are used which are capable to deduce patterns given by the training dataset in order to create a generative model for classification. The generative model will be able to classify unseen data sample based on its content to its respective class. In the field of malware detection, machine learning techniques have been recognized to be effective to detect unseen variants of malwares [11]. One of the limiting factors of machine learning algorithms is the false rate which depends on multitude of factors such as irrelevant features, ineffective selection of features, or incompetent classifier algorithms [11]. In addition, irrelevant features of the training dataset will increase the number of features in the feature space which will lead to increase the time needed to create a generative model for classification. Also, the increasing number of features will require more computing resource. Hence, informative textual string attributes are employed as the features in this project with the aim to increase the classifier accuracy and to decrease the computational overhead.

1.3 Objectives

The purpose of this study is to accomplish the following objectives:

1. To develop machine learning classifiers which are constructed by using minimum number of textual features extracted from both malicious and benign files in order to differentiate between self-mutated and non-mutated files.

2. To identify the most suitable machine learning classifier model based on its accuracy level and false positive rate.
3. To compare the outcomes of generated machine learning classifiers with conventional anti-malware software based on signature matching technique.

1.4 Scope of the research

The concentration of this research will be on the detection of new obfuscated malicious files of the metamorphic type. The primary goal is to enhance the accuracy level of machine learning classifier models compared to the conventional malware detection. Owing to the exploratory identity of this research, the collected sets of both training and testing stages of machine learning classifier models are from offline traffic. The scopes of this study are stated as follows:

1. Training dataset comprised of benign and malware is gathered from different online samples providers. The collected dataset is utilized to develop generative classifier model.
2. Data collection, preprocessing, machine learning training and classification were conducted in virtual environment for the sake of protecting workstation from getting infected by collected malwares.
3. Textual strings were chosen as a feature for generating machine learning models. The textual strings were extracted from dataset executables using disassembler software.
4. Multiple machine learning classifier models were examined in order to determine the best classifier in terms of accuracy of detection and false rates.

1.5 Thesis organization

The report consists of a total number of five chapters which are as follows. Chapter 2 describes the literature review of related works. Chapter 3 presents the methodology followed to conduct the research project. Chapter 4 presents the experimental findings and their analysis in the which the obtained results are compared

against the traditional signature-based technique. Chapter 5 presents the concluding remarks and the recommendations for future work.

REFERENCES

1. Canali, D., Lanzi, A., Balzarotti, D., Kruegel, C., Christodorescu, M., and Kirda, E. (2012). *A quantitative study of accuracy in system call-based malware detection*. Paper presented at *the Proceedings of the 2012 International Symposium on Software Testing and Analysis*, 122-132.

2. Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S. and Weaver, N. (2003). *Inside the slammer worm*. *IEEE Security & Privacy*. 1(4), 33–39.

3. Mcafee.com. (2014). [online] Available at: https://www.mcafee.com/enterprise/en-us/assets/executive-summaries/es-economic-impact-cybercrime.pdf [Accessed 25 march. 2019].

4. kumar Sasidharan, S., & Thomas, C. (2018, September). *A Survey on Metamorphic Malware Detection based on Hidden Markov Model*. In 2018 *International Conference on Advances in Computing, Communications and Informatics* (ICACCI) (pp. 357-362). IEEE.

5. Jain, S., and Meena, Y. K. (2011). *Byte Level n-Gram Analysis for Malware Detection*. *Computer Networks and Intelligent Computing*, 157, 51-59.

6. You, I. and Yim, K. (2010) *Malware Obfuscation Techniques: A Brief Survey*. *Proceedings of International conference on Broadband, Wireless Computing*, Communication and Applications, Fukuoka, 4-6 November 2010, 297-300.

7. Egele, M., Scholte, T., Kirda, E. and Kruegel, C. (2012) *A Survey on Automated Dynamic Malware-Analysis Techniques and Tools*. *Journal in ACM Computing Surveys*, 44, Article No.6.

8. Shalaginov, A., Banin, S., Dehghantanha, A., & Franke, K. (2018). *Machine learning aided static malware analysis: A survey and tutorial*. *Cyber Threat Intelligence*, 7-45.

9. El Merabet, H., & Hajraoui, A. (2019). *A Survey of Malware Detection Techniques based on Machine Learning. International Journal of Advanced Computer Science and Applications*, 10(1), 366-373.

10. Sharma, A., & Sahay, S. K. (2014). *Evolution and detection of polymorphic and metamorphic malwares: A survey*. *arXiv preprint* arXiv:1406.7061.

11. Vinod, P., Laxmi, V. and Gaur, M. S. (2012). *REFORM: relevant features for malware analysis*. *In 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA '12)*. IEEE, 738–744.

12. Wong, W., & Stamp, M. (2006). *Hunting for metamorphic engines. Journal in Computer Virology*, 2(3), 211-229.

13. S. Alam, R. N. Horspool, I. Traore, and I. Sogukpinar, *A framework for metamorphic malware analysis and real-time detection, Computers & Security*, vol. 48, pp. 212–233, 2015.

14. Mahawer, Devendra Kumar, and A. Nagaraju. *Metamorphic malware detection using base malware identification approach. Security and Communication Networks 7*, no. 11 (2014): 1719-1733.

15. VMware workstation, https://www.vmware.com/my/products/workstation-player/workstation-player-evaluation.html [Retrieved on 20 March 2019]

16. malshare, https://malshare.com, [Retrieved on 20 March 2019]

17. Dasmalwer, https://dasmalwerk.eu/, [Retrieved on 20 March 2019]

18. IDA pro, https://www.hex-rays.com/products/ida/, [Retrieved on 20 March 2019]

19. WEKA, https://www.cs.waikato.ac.nz/ml/weka/, [Retrieved on 20 March 2019]

20. NGVCK, vxheaven.org, [Retrieved on 20 March 2019]

21. WinMerge, http://winmerge.org/downloads/?lang=en, [Retrieved on 20 March 2019]

22. nirsoft, https://nirsoft.net, [Retrieved on 20 March 2019]

23. Walenstein, Andrew, Michael Venable, Matthew Hayes, Christopher Thompson, and Arun Lakhotia. *Exploiting similarity between variants to defeat malware*. In Proc. BlackHat DC Conf. 2007.

24. Ponnambalam, P. *Measuring malware evolution*. Master's Thesis. San Jose State University, 2015.

25. Highland, Harold Joseph. *A history of computer viruses—Introduction*. (1997): 412-415.

26. Smith, Craig, Ashraf Matrawy, Stanley Chow, and Bassem Abdelaziz. *Computer worms: Architectures, evasion strategies, and detection mechanisms*. *Journal of Information Assurance and Security* 4 (2009): 69-83.

27. Front Matter. Minerva, 1973. 11(3). URL http://www.jstor.org/ stable/41820151.

28. Sriramachandramurthy, Rajendran, Siva K. Balasubramanian, and Monica Alexandra Hodis. *Spyware and adware: how do internet users defend themselves?. American Journal of Business 24*, no. 2 (2009): 41-52.

29. Mohaisen, Abedelaziz, and Omar Alrawi. *Unveiling zeus: automated classification of malware samples. In Proceedings of the 22nd International Conference on World Wide Web*, pp. 829-832. ACM, 2013.

30. Bunten, Andreas. *Unix and linux based rootkits techniques and countermeasures. In 16th Annual First Conference on Computer Security Incident Handling*, Budapest. 2004.

31. Islam, Md Nazmul, and Sandip Kundu. *Pmu-trojan: on exploiting power management side channel for information leakage. In Proceedings of the 23rd Asia and South Pacific Design Automation Conference*, pp. 709-714. IEEE Press, 2018.

32. Sreenivas, R. Sreeram, and R. Anitha. *Detecting keyloggers based on traffic analysis with periodic behaviour. Network Security 2011*, no. 7 (2011): 14-19.

33. Vinod, P., R. Jaipur, V. Laxmi, and M. Gaur. *Survey on malware detection methods. In Proceedings of the 3rd Hackers' Workshop on computer and internet security* (IITKHACK'09), pp. 74-79. 2009.

34. Egele, Manuel, Theodoor Scholte, Engin Kirda, and Christopher Kruegel. *A survey on automated dynamic malware-analysis techniques and tools. ACM computing surveys* (CSUR) 44, no. 2 (2012): 6.

35. Bailey, Michael, Jon Oberheide, Jon Andersen, Z. Morley Mao, Farnam Jahanian, and Jose Nazario. *A survey on automated dynamic malware-analysis*

techniques and tools. ACM computing surveys, pp. 178-197. Springer, Berlin, Heidelberg, 2007.

36. Lee, Tony, Jigar Mody, Ying Lin, Adrian Marinescu, and Alexey Polyakov. *Application behavioral classification*. U.S. Patent Application 11/608,625, filed June 14, 2007.

37. Zhong, Yang, Hirofumi Yamaki, and Hiroki Takakura. A malware *Classification method based on similarity of function structure. In 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, pp. 256-261. IEEE, 2012.

38. Canzanese, Raymond, Moshe Kam, and Spiros Mancoridis. *Toward an automatic, online behavioral malware classification system. In 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*, pp. 111-120. IEEE, 2013.

39. Beaucamps, P., Gnaedig, I., and Marion, J. Y. (2010). *Behavior Abstraction in Malware Analysis. Runtime Verification*, 6418, 168-182.

40. Pfeffer, A., Call, C., Chamberlain, J., Kellogg, L., Ouellette, J., Patten, T., et al. (2012). *Malware analysis and attribution using genetic information*. Paper presented at the Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on, 39-45.

41. White, J. S., Pape, S. R., Meily, A. T., and Gloo, R. M. (2013). *Dynamic Malware Analysis Using IntroVirt: a Modified Hypervisor-Based System. Cyber Sensing* 2013, 8757.

42. Vasudevan, A., and Yerraballi, R. (2006). *SPiKE: engineering malware analysis tools using unobtrusive binary-instrumentation. Paper presented at the Proceedings of the 29th Australasian Computer Science Conference* - Volume 48.

43. Vinod, P., Laxmi, V., Gaur, M. S., Naval, S., and Faruki, P. (2013). *MCF: MultiComponent Features for Malware analysis. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications Workshops* (Waina), 1076-1081.

44. Cifuentes, C., and Fraboulet, A. (1997). *Intraprocedural static slicing of binary executables*, 188.

45. Ghiasi, M., Sami, A., and Salehi, Z. (2012). *Dynamic Malware Detection Using Registers Values Set Analysis. 2012 9th International Isc Conference on Information Security and Cryptology* (Iscisc), 54-59.

46. Christodorescu, M., Jha, S., Seshia, S. A., Song, D., and Bryant, R. E. (2005a, 8-11 May 2005). *Semantics-aware malware detection. Paper presented at the Security and Privacy, 2005 IEEE Symposium* on, 32-46.

47. Idika, N., and Mathur, A. P. (2007). *A Survey of Malware Detection Techniques*,Department of Computer Science,Purdue University, West Lafayette, IN 47907.USAo. Document Number)

48. Weaver, N., Paxson, V., Staniford, S., and Cunningham, R. (2003). *A taxonomy of computer worms. Paper presented at the Proceedings of the 2003 ACM workshop on Rapid malcode*. from http://dl.acm.org/citation.cfm?doid=948187.948190

49. Vinod, P., Laxmi, V., and Gaur, M. S. (2011). *Scattered Feature Space for Malware Analysis. Advances in Computing and Communications*, Pt I, 190, 562-571.

50. Moser, A., Kruegel, C., and Kirda, E. (2007). *Limits of static analysis for malware detection. Paper presented at the Computer Security Applications Conference*, 2007. ACSAC 2007. Twenty-Third Annual, 421-430.

51. Alam, S., Horspool, R. N., Traore, I. and Sogukpinar, I. (2015). *A framework for metamorphic malware analysis and real-time detection. Computers & Security*. 48, 212–233.

52. LeDoux, C. and Lakhotia, A. (2015). *Malware and machine learning. In Intelligent Methods for Cyber Warfare*. Springer, 1–42.

53. Canfora, G., Iannaccone, A. N. and Visaggio, C. A. (2013). *Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics. Journal of Computer Virology and Hacking Techniques*, 1–17.

54. Attaluri, S., McGhee, S. and Stamp, M. (2009). *Profile hidden markov models and metamorphic virus detection. Journal in Computer Virology*. 5(2), 151–169.

55. Shanmugam, G., Low, R. M. and Stamp, M. (2013). *Simple substitution distance and metamorphic detection. Journal of Computer Virology and Hacking Techniques*. 9(3), 159–170.

56. ong, F. and Touili, T. (2012). *Efficient malware detection using model-checking*. In FM 2012: Formal Methods. (pp. 418–433). Springer.

57. Rad, B. B., Masrom, M. and Ibrahim, S. (2012). *Opcodes histogram for classifying metamorphic portable executables malware. In International Conference on eLearning and e-Technologies in Education (ICEEE)*. IEEE, 209–213.

58. Kenan, Z. and Baolin, Y. *Malware Behavior Classification Approach Based on Naive Bayes*. 2012. 7: 203–210.

59. Hu, W. and Tan, Y. *On the robustness of machine learning based malware detection algorithms. 2017 International Joint Conference on Neural Networks (IJCNN)*. 2017. 1435–1441.

60. Gumus, F., Sakar, C. O., Erdem, Z. and Kursun, O. *Online Naive Bayes classification for network intrusion detection. 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*. 2014. 670–674.

61. Gupta, D., & Rani, R. (2018). *Big Data Framework for ZeroDay Malware Detection. Cybernetics and Systems*, 49(2), 103-121.

62. Cho, I. K., Kim, T. G., Shim, Y. J., Ryu, M., & Im, E. G. (2016). *Malware Analysis and Classification Using Sequence Alignments. Intelligent Automation & Soft Computing*, 22(3), 371- 377.

63. Burnap, P., French, R., Turner, F., & Jones, K. (2018). *Malware classification using self organising feature maps and machine activity data. computers & security, 73*, 399-410

64. AlAhmadi, B. A., & Martinovic, I. (2018, May). *MalClassifier: Malware family classification using network flow sequence behaviour. In APWG Symposium on Electronic Crime Research (eCrime)*, 2018 (pp. 1-13). IEEE.

65. Nari, S., & Ghorbani, A. A. (2013, January). *Automated malware classification based on network behavior. In 2013 International Conference on Computing, Networking and Communications (ICNC)* (pp. 642-647). IEEE.

66. Gandotra, E., Bansal, D., & Sofat, S. (2014, September). *Integrated framework for classification of malwares. In Proceedings of the 7th International Conference on Security of Information and Networks* (p. 417). ACM

67. Islam, R., Tian, R., Batten, L. M., & Versteeg, S. (2013). *Classification of malware based on integrated static and dynamic features. Journal of Network and Computer Applications*, 36(2), 646-656.

68. Tian, R., Batten, L., Islam, R., & Versteeg, S. (2009, October). *An automated classification system based on the strings of trojan and virus families. In Malicious and Unwanted Software (MALWARE)*, 2009 4th International Conference on (pp. 23-30). IEEE.

69. Khammas, B. M., Monemi, A., Bassi, J. S., Ismail, I., Nor, S. M., & Marsono, M. N. (2015). *Feature selection and machine learning classification for malware detection. Jurnal Teknologi*, 77(1).

70. BitShred, https://github.com/dbrumley/bitshred.

71. Uppal, Dolly, Vishakha Mehra, and Vinod Verma. *Basic survey on malware analysis, tools and techniques. International Journal on Computational Sciences & Applications (IJCSA)* 4.1 (2014): 103.

72. Khammas, b., 2017. *Network level malware detection based on packet payload classification*. Ph.d. UTM.