FPGA IMPLEMENTATION OF NAIVE BAYES CLASSIFIER FOR NETWORK SECURITY

AHMAD ZULZHAFRI BIN MOHAMAD ZUKI

UNIVERSITI TEKNOLOGI MALAYSIA

FPGA IMPLEMENTATION OF NAIVE BAYES CLASSIFIER FOR NETWORK SECURITY

AHMAD ZULZHAFRI BIN MOHAMAD ZUKI

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic Systems)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2018

To my beloved family members who always there for me,

my friends who assists, accompanying me now and then,

and also to my supervisor who guide me through the research's hardships

ACKNOWLEDGEMENT

I am delighted to express my gratitude to Allah for giving me this opportunity to proceed with this project. I would also like to express my gratitude to my supervisor, Dr. Nadzir Marsono for helping and giving lots of support and guidance for me while finishing this project. Without her guidance and support, I would not be able to complete this project on time.

In addition, I would like to give a big thanks to my dearest family members for giving me strength throughout this project. Their pure blessings were the greatest motivation for me to proceed with this project and to face and to solve all problems in this project.

My fellow course mates should also be recognized for their supports and motivation in completing this project. Their tips and advices always help me to solve problems that occur throughout this project. I am also grateful to Universiti Teknologi Malaysia (UTM) for providing such a nice facility in order for me to do this project in comfort.

ABSTRACT

In the vast usage of internet nowadays, the rate of cybercrime such as fraud, hacking, identity theft, network intrusion, software piracy and espionage are becoming more critical. Malware code writers used this chance to create malware that able to breach the security and gain access to the information. Hence, the importance of malware detection system becoming more significant as the users need the protection from the malware threats. Most of malware detection systems implement signature based classification where only known malware can be detected. Nowadays, new malwares are able to change its signature sequence regularly in order to avoid detection. This polymorphic malware becomes the limitation for signature based detection approach. This project aim is to proposed signature-based detection approach that able to detect polymorphic malware by using Naïve Bayes algorithm. The integration of the classifier architecture onto FPGA board in order to measures the performances of the system. The feature from network traffic subset to Snort signature detection of known malware and benign samples are extracted using overlapping N-gram string format. The data set is then being used for training and testing for the classifier. The classifier for the malware detection used Naïve Bayes algorithm that using Bayesian Theorem probability for the features in the data set to determine types of the flow. The model is then being implemented into hardware FPGA architecture and being coded in RTL. The target FPGA that being used in Vivado software is Xilinx Virtex-7 VC709 that able to support the system requirements. The hardware performance of the model was analyzed and compared with the Naïve Bayes software classifier for the performance evaluation. The proposed hardware NB malware detection classifier has managed to achieve 96.3% accuracy and improved FPR rate of 3.1%. The hardware NB malware detection classifier on FPGA architecture also able to achieve better resource utilization and improved detection speed of 0.13 μs per flow.

ABSTRAK

Di zaman penggunaan internet yang meluas, jenayah siber turut juga menjadi jenayah utama yang ,mampu untuk meruntuhkan ekonomi negara. Melalui internet, transaksi-transaksi penting seperti pemindahan wang, transaksi bank, informasi peribadi, data sulit negara dan informasi-informasi penting dilakukan setaip hari. Pencipta virus komputer menggunakan peluang ini untuk mencuri data-data penting tersebut demi kepentingan mereka. Pengguna perlulah lebih peka terhadap jenayah siber ini. Kepentingan anti virus semakin meningkat bertujuan untuk memberikan perlindungan informasi kepada pengguna. Kaedah konvensional untuk mengesan virus ini hanya mampu mengesan virus yang mempunyai ciri-ciri program yang sudah dikenalpasti oleh pengesan virus. Di zaman sekarang, virus baru mampu untuk mengubah ciri-ciri lantas menyebabkan pengesan virus gagal mengasingkan virus. Projek ini bertujuan untuk mencipta model pengesan virus yang dilatih menggunakan contoh virus yang telah dikenalpasti dengan menggunakan ciri-ciri pengesahan Snort. Ciri-ciri tersebut akan melalui beberapa proses seperti pengurangan ciri dan dapatan informasi fasa sebelum berubah kepada format pertindanan n-gram. Algorithma yang digunakan oleh model tersebut ialah Naive Bayes (NB) yang berpandukan kepada teori Bayesian. Kebolehan mengadaptasikan algorithma tersebut kepada model tersebut adalah model itu mampu untuk diprogram untuk mempelajari ciri-ciri virus yang baru. Pengkalan data yang sudah dilatih dan mempunyai ciri-ciri penting virus akan melalui simulasi menggunakan perisian WEKA bagi mendapatkan prestasi ketepatan simulasi perisian. Selepas itu, model tersebut akan dibina menggunakan perkakasan FPGA. Prestasi yang dicapai oleh model perkakasan tersebut akan dibandingkan dengan prestasi model perisian. Kedua-dua model virus pengesan menggunakan NB algoritma tersebut akan dinilai dari segi ketepatan pengklasifikasian, masa pemprosesan dan beberapa aspek lain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	-	Artificial Neural Network
ASIC	-	Application-specific Integrated Circuit
API	-	Application Program Interface
CNG	-	Common n-gram
CPU	-	Central Processing Unit
DLL	-	Dynamic-link Library
FNR	-	False Negative Rate
FPGA	-	Field Programmable Gate Array
FPR	-	False Positive Rate
HDL	-	Hardware Description Language
IBK	-	Instance Based Learner
IG	-	Information Gain
LUT	-	Look-up Table
NB	-	Naïve Bayes
RTL	-	Register Transfer Level
SVM	-	Support Vector Machine
TCL	-	Tool Command Language
TCP	-	Transmission Control Protocol
TNR	-	True Negative Rate
TPR	-	True Positive Rate

CHAPTER 1

INTRODUCTION

1.1 Problem Background

Malware also known as malicious software is any computer program that bring harm to the computer hosts. Malware is the term used to represents viruses, Trojan horse, Backdoors, worms and etc. Most common actions from the malicious code is replicates rapidly by infecting any host files or any system inside the computer. It also capable of multiplying countless time to form new obfuscated virus code [1]. Besides, it is easier to learn on how to create malware as there are high resources of malware construction and attacking tools in the internet. Hackers also able to buy malware on the black market and learn on anti-detection techniques used to create undetectable malware. Resulting from that, anyone can learn easily to become hacker and not depending on programming skills anymore. In [2], it states that there are major malware attacked are being created by script-kiddies in the recent years.

The classification to identify malware is that any software or system that performing malicious action can be considered as malware. The crimes resulting from malware are espionage, identity stealing, information breach, and lots more. Anti-virus scanners are not able to scan and filter out most of malware as the increasing diversity and new malware are introduced every day that resulting in millions of users being hacked. Anti-viruses can't provide perfection protection towards malware and based on statistic, 6 563 145 were attacked and 4 000 000 malware objects are detected in 2015 only. The data provided by Kaspersky Labs (2016). By 2019, Jupiter Research (2016) predicts that the cost for the data breach resulting from malware will increases significantly to $2.1 trillion globally.

Malware protection for computer system is significantly important in order to secure the data regardless for business purpose or single user. This cybersecurity is important to prevent data leaked and espionage. The need for accurate and reliable malware detection methods is high to overcome the frequent attacks. Conventional detection using static and dynamic methods does not able to provide efficient detection as the malware keep changing in time. Machine learning techniques is suitable to overcome the limitation of the malware detection.

This paper discussing the implementation of machine learning using Naïve Bayes classifier embedded in FPGA to increase the efficiency, reliability, accuracy, throughput and computational resources [3]. To generate the detection models for the malware, signatures from known malware samples are used in the algorithm for classic malware detection. The signatures are created by determining the unique fingerprint pattern for each malware family. It can produce accurate classifier with less low false rate. In spite of that, the detection method not able to detect new malicious samples as the code are more complicated and can slipped through the detection using packaging and obfuscated techniques. To prevent the malware undetected rate, the system can increase the malware database by keep updating the malware samples frequently. There is a draw back when increasing the database samples that resulting in increasing scanning time as the database increase in size [4]. Thus, the performance will reduce in term of throughput.

Data mining is used in this paper to overcome the limitation of the signature based classifier mention above. In this paper also, the implementation of Naïve Bayes classifier for the malware detection uses the concept of supervised classifier machine learning. Based on supervised classifier, the samples of known malware and benign program are used as the training set. Then, the feature of known malware and benign samples are extracted and used as classification model to classify types of malware and benign classes.

The samples are transform to n-gram sequences by using feature reduction process [5]. N-gram sequences are usually used in several types of classifier algorithm such as Naïve Bayes, Random Forest and Decision Tree classifier. In the purposed methodology, the n-grams sequences are extracted from the known benign and malware samples by using feature reduction method. The n-grams will assist as the class for the Naïve Bayes classifier to scan through unknown program samples. Classwise document frequency also being used in the flow to extract prominent features that able to differentiate malware from benign programs. Classwise document

frequency condense the huge feature size that will reduce the scanning time for that classifier during testing and training phase.

For finding the suitable number of n-grams, simulations using several types of classifier such as Naïve Bayes, Random Forest, J48, Instance Based Learner (IBK) and AdaBoost1 which are supported in WEKA tools [6]. The simulations are tested using the same training and testing data set. Based on the results, the proposed classifier NB produce high accuracy with slight false rate but the implementation will still use NB as the classifier due to advantages of using NB classifier that is more suitable to be used when the training and testing data set are small compared to other classifier that require bigger database for higher accuracy. The aim for this project is to focusing in the hardware implementation where hardware performance is more significant compared to reliability. Choosing NB classifier will slightly reduce the accuracy and reliability [7] of the classification but the hardware performances are believed to be better when implemented inside FPGA. The implementation of the classifier algorithm into FPGA will be done in the next phase. The target device is Xilinx Virtex-6 LX760 FPGA. The algorithm is transform into RTL code by using Verilog programming language [8]. The software used in the hardware design are Vivado that will virtually connect the algorithm in RTL code into the FPGA board while the software also used to create the testbench for the whole hardware system. The computational resources and performance of the system is observed and compared to other hardware classifier.

Simulation using Matlab [9] also being done onto the selected NB classifier algorithm to generate new results that will be compared with the results obtain from the hardware simulation. The results should be comparable relatively the same. This project will be focusing more on hardware simulation as the classifier algorithm already being set as NB classifier. Improvement and modification will be done in hardware parts to increase the performance, throughput, accuracy, reliability and processing time. Slight trade off in accuracy and hardware performance might be seen as only one of the aspect can be improve while the other will degrade.

1.2 Problem Statement

Most malware scanners employ signature based detection methods and these scanners fail to detect unseen and obfuscated malware samples. Malware has becoming harder to detect as new malware are being introduce frequently. The existing malware also has the capability to replicate and change its code in order to avoid anti-virus and detection. Exact pattern matching does not suitable to detect new malware as the pattern matching uses limited sample databases. This disadvantage is the reason for the increasing data breach although new anti-viruses are being sold in the market.

Conventional malware detection is not suitable to implement deep machine learning technique due to restricted programmable ability of the normal system. The existing systems in market only implement several hardware accelerators such as pipelining, parallel classifications, fixed point data representation and look-up table. However, most of these hardware implementation works are designed for others application classification purpose instead of malware detection classification. The limited computational resources in normal malware detection system exhibit the throughput of the detection system. FPGAs are more suitable for implementing machine learning as they can handle different algorithms in computing, logic, and memory resources in the same device. Besides that, the performance of FPGAs is much faster compared to other chip as users can hard code the operations and algorithms into the hardware.

1.3 Objective

The aim of this project is to achieve the following objective:

1. To implement signature based approach for detecting malicious code by using n–grams which extracted from TCP traffic of benign and malware samples.
2. To obtained better performance on CPU usage by using Naive Bayes (NB) Classifier onto hardware (FPGA) based using machine learning to detect malware.

1.4 Scope

For this project, the scopes covered are as follows:

- The implementation of the malware detection using machine learning are done at the hardware based (middle box) where it is assumed the data packet (stateless) contains payload in hex in form of overlapping n-gram.
- Focusing on implementation of Naïve Bayes classifier to detect malware onto FPGA architecture.
- Using the existing Naïve Bayes classifier algorithm while the features are reduce based on Snort Signatures
- The system will function as malware detection system that implement NB in determining the class of malware either malware or benign.

1.5 Organization

The report is organized with 5 chapters. Chapter 1 provide the introduction of the project including background, problem statement, objective, and scope. Chapter 2 reviews the related literature review on state-of-the-art of N-gram text classification, malware types, malware signature, Naive Bayes algorithm, FPGA implementation using Naive Bayes classifier, limitation, research gap and other related works. Chapter 3 describe the research methodology for the overview of proposed work, research activities, techniques and tools used. Chapter 4 elaborates the design implementation onto FPGA hardware as hardware accelerator and the evaluation on proposed malware detection Naive Bayes classifier. Chapter 5 summarize the proposed works and discussed on the future works.

REFERENCES

1. Szor, P. *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional. 2005.
2. Koniaris, I., Papadimitriou, G., Nicopolitidis, P. and Obaidat, M. Honeypots deployment for the analysis and visualization of malware activity and malicious connections. *2014 IEEE International Conference on Communications (ICC)*. 2014. 1819–1824.
3. Choi, S.-W. and Ho Lee, C. A FPGA-based parallel semi-naive Bayes classifier implementation. 2013. 10: 20130673–20130673.
4. Zhang, X., Ramachandran, A., Zhuge, C., He, D., Zuo, W., Cheng, Z., Rupnow, K. and Chen, D. Machine learning on FPGAs to face the IoT revolution. *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 2017. 819–826.
5. Liangboonprakong, C. and Sornil, O. Classification of malware families based on N-grams sequential pattern features. *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*. 2013. 777–782.
6. Witten, I. H., Frank, E. and Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 2011. ISBN 0123748569, 9780123748560.
7. Venugopal, K. R. and Patnaik, L. M. *Computer Networks and Intelligent Computing: 5th International Conference on Information Processing, ICIP 2011, Bangalore, India, August 5-7, 2011. ... In Computer and Information Science)*. 1st ed. Springer Publishing Company, Incorporated. 2011.
8. Meng, H., Appiah, K., Hunter, A. and Dickinson, P. FPGA implementation of Naive Bayes classifier for visual object recognition. *CVPR 2011 WORKSHOPS*. 2011. 123–128.
9. Zhang, H. The Optimality of Naive Bayes, 2004.
10. Highland, H. J. A history of computer viruses – Introduction. *Computers and Security*, 1997. 16(5): 412 – 415.

11. Smith, C., Matrawy, A., Chow, S. and Abdelaziz, B. Computer Worms: Architectures, Evasion Strategies, and Detection Mechanisms. 2009. 4: 69–83.

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

13. Yilmaz, S. and Zavrak, S. Adware: A Review. 2015. 6: 5599–5604.

14. Mohaisen, A. and Alrawi, O. Unveiling Zeus: Automated Classification of Malware Samples. *Proceedings of the 22Nd International Conference on World Wide Web*. New York, NY, USA: ACM. 2013, WWW '13 Companion. 829–832.

15. Bunten, A. UNIX and Linux based Rootkits Techniques and Countermeasures. 2004.

16. Islam, M. N. and Kundu, S. PMU-Trojan: On Exploiting Power Management Side Channel for Information Leakage. *Proceedings of the 23rd Asia and South Pacific Design Automation Conference*. Piscataway, NJ, USA: IEEE Press. 2018, ASPDAC '18. 709–714.

17. Sreenivas, R. S. and Anitha, R. Detecting keyloggers based on traffic analysis with periodic behaviour. *Network Security*, 2011. 2011: 14–19.

18. Miramirkhani, N., Appini, M. P., Nikiforakis, N. and Polychronakis, M. Spotless Sandboxes: Evading Malware Analysis Systems Using Wear-and-Tear Artifacts. *2017 IEEE Symposium on Security and Privacy (SP)*. 2017. 1009–1024.

19. Moser, A., Kruegel, C. and Kirda, E. Limits of Static Analysis for Malware Detection. *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. 2007. 421–430.

20. Christodorescu, M. and Jha, S. Static Analysis of Executables to Detect Malicious Patterns. *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12*. Berkeley, CA, USA: USENIX Association. 2003, SSYM'03. 12–12.

21. Kolosnjaji, B., Eraisha, G., Webster, G., Zarras, A. and Eckert, C. Empowering convolutional networks for malware classification and analysis. *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017. 3838–3845.

22. Egele, M., Scholte, T., Kirda, E. and Kruegel, C. A Survey on Automated Dynamic Malware-analysis Techniques and Tools. *ACM Comput. Surv.*, 2008.

44(2): 6:1–6:42.

23. Liu, W., Ren, P., Liu, K. and Duan, H. Behavior-Based Malware Analysis and Detection. 2011.

24. Yoo, I. S. and Ultes-Nitsche, U. Non-signature based virus detection. *Journal in Computer Virology*, 2006. 2(3): 163–186.

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

26. Mori, A. Detecting Unknown Computer Viruses - A New Approach, 2003.

27. Reddy, D. K. S. and Pujari, A. K. N-gram analysis for computer virus detection. *Journal in Computer Virology*, 2006. 2(3): 231–239.

28. Lever, C., Kotzias, P., Balzarotti, D., Caballero, J. and Antonakakis, M. A Lustrum of Malware Network Communication: Evolution and Insights. *2017 IEEE Symposium on Security and Privacy (SP)*. 2017. 788–804.

29. Cakir, B. and Dogdu, E. Malware Classification Using Deep Learning Methods. *Proceedings of the ACMSE 2018 Conference*. New York, NY, USA: ACM. 2018, ACMSE '18. 10:1–10:5.

30. 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.

31. 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.

32. Ozsoy, M., Khasawneh, K. N., Donovick, C., Gorelik, I., Abu-Ghazaleh, N. and Ponomarev, D. Hardware-Based Malware Detection Using Low-Level Architectural Features. *IEEE Transactions on Computers*, 2016. 65(11): 3332–3344.

33. Shankar, V. G., Somani, G., Gaur, M. S., Laxmi, V. and Conti, M. AndroTaint: An efficient android malware detection framework using dynamic taint analysis. *2017 ISEA Asia Security and Privacy (ISEASP)*. 2017. 1–13.

34. Baldangombo, U., Jambaljav, N. and Horng, S. A Static Malware Detection System Using Data Mining Methods. *CoRR*, 2013. abs/1308.2831. URL `http://arxiv.org/abs/1308.2831`.

35. Baskaran, B. and Ralescu, A. L. A Study of Android Malware Detection Techniques and Machine Learning. *MAICS*. 2016.

36. Kephart, J. O. A Biologically Inspired Immune System for Computers. *In Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press. 1994. 130–139.

37. Griffin, K., Schneider, S., Hu, X. and Chiueh, T.-C. Automatic Generation of String Signatures for Malware Detection. *Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection*. Berlin, Heidelberg: Springer-Verlag. 2009, RAID '09. 101–120.

38. Abou-Assaleh, T., Cercone, N., Keselj, V. and Sweidan, R. Detection of New Malicious Code Using N-grams Signatures., 2004.

39. Kolter, J. Z. and Maloof, M. A. Learning to Detect Malicious Executables in the Wild. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM. 2004, KDD '04. 470–478.

40. Eagle, C. *The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler*. San Francisco, CA, USA: No Starch Press. 2008. ISBN 1593271786, 9781593271787.

41. Kolter, J. Z. and Maloof, M. A. Learning to Detect and Classify Malicious Executables in the Wild. *J. Mach. Learn. Res.*, 2006. 7: 2721–2744.

42. Schultz, M. G., Eskin, E., Zadok, F. and Stolfo, S. J. Data mining methods for detection of new malicious executables. *Proceedings 2001 IEEE Symposium on Security and Privacy. S P 2001*. 2001. 38–49.

43. Kephart, J. O., Sorkin, G. B., Arnold, W. C., Chess, D. M., Tesauro, G. J. and White, S. R. Biologically Inspired Defenses Against Computer Viruses. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 1995, IJCAI'95. 985–996.

44. Yang, Y. and Pedersen, J. O. A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 1997, ICML '97. 412–420.

45. GavriluÅ£, D., Cimpoesu, M., Anton, D. and Ciortuz, L. Malware detection using machine learning. *2009 International Multiconference on Computer Science and Information Technology*. 2009. 735–741.

46. Singhal, P. and Raul, N. Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks. *CoRR*, 2012. abs/1205.3062.

47. Alazab, M., Venkatraman, S., Watters, P. and Alazab, M. Zero-day Malware Detection Based on Supervised Learning Algorithms of API Call Signatures. *Proceedings of the Ninth Australasian Data Mining Conference - Volume 121*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc. 2011, AusDM '11. 171–182.

48. Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. 2006.

49. Biau, G. Analysis of a Random Forests Model. *J. Mach. Learn. Res.*, 2012. 13: 1063–1095.

50. Chumachenko, K. Machine Learning Methods for Malware Detection and Classification., 2017.

51. Appiah, K., Hunter, A., Dickinson, P. and Meng, H. Binary object recognition system on FPGA with bSOM. *23rd IEEE International SOC Conference*. 2010. 254–259.

52. Wang, T. and Xu, N. Malware variants detection based on opcode image recognition in small training set. *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. 2017. 328–332.

53. Salih, M. H. and Arshad, M. R. Embedded Parallel Systolic architecture for multi-filtering techniques using FPGA. *2010 2nd International Conference on Electronic Computer Technology*. 2010. 122–127.

54. Torres-Huitzil, C. and Arias-Estrada, M. FPGA-Based Configurable Systolic Architecture for Window-Based Image Processing. *EURASIP Journal on Advances in Signal Processing*, 2005. 2005(7): 264713.

55. Rish, I. An Empirical Study of the NaÃ¯ve Bayes Classifier. 2001. 3.

56. Gao, C. and Lu, S.-L. Novel FPGA based Haar classifier face detection algorithm acceleration. *2008 International Conference on Field Programmable Logic and Applications*. 2008. 373–378.

57. Azarian, A. and Cardoso, J. Pipelining Data-Dependent Tasks in FPGA-based Multicore Architectures. 2016: –.

58. Cho, J., Mirzaei, S., Oberg, J. and Kastner, R. Fpga-based Face Detection System Using Haar Classifiers. *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. New York, NY, USA: ACM. 2009, FPGA '09. 103–112.

59. Langseth, H. and Nielsen, T. D. Classification using Hierarchical Naïve Bayes models. *Machine Learning*, 2006. 63(2): 135–159.

60. Athow, J. L. and Al-Khalili, A. J. Implementation of large-integer hardware multiplier in Xilinx FPGA. *2008 15th IEEE International Conference on Electronics, Circuits and Systems*. 2008. 1300–1303.

61. Santos, I., Penya, Y. K., Devesa, J. and Bringas, P. G. N-grams-based File Signatures for Malware Detection. *ICEIS*. 2009.

62. Marsono, M. N., El-Kharashi, M. W. and Gebali, F. Binary LNS-based naive Bayes inference engine for spam control: noise analysis and FPGA implementation. *IET Computers Digital Techniques*, 2008. 2(1): 56–62.

63. Ismail, I., Nor, S. M. and Marsono, M. N. Stateless Malware Packet Detection by Incorporating Naive Bayes with Known Malware Signatures. *Appl. Comp. Intell. Soft Comput.*, 2014. 2014: 5:5–5:5.

64. Caswell, B., Foster, J. C., Russell, R., Beale, J. and Posluns, J. *Snort 2.0 Intrusion Detection*. Syngress Publishing. 2003.

65. Kolter, J. Z. and Maloof, M. A. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, 2006. 7(Dec): 2721–2744.

66. Moskovitch, R., Stopel, D., Feher, C., Nissim, N. and Elovici, Y. Unknown malcode detection via text categorization and the imbalance problem. *2008 IEEE International Conference on Intelligence and Security Informatics*. 2008. 156–161.

67. Abou-Assaleh, T., Cercone, N., Keselj, V. and Sweidan, R. Detection of New Malicious Code Using N-grams Signatures., 2004.

68. Ismail, I., Marsono, M. N. and Nor, S. M. Detecting Worms Using Data Mining Techniques: Learning in the Presence of Class Noise. *2010 Sixth International Conference on Signal-Image Technology and Internet Based Systems*. 2010. 187–194.