N-GRAM FEATURE EXTRACTION AND NAÏVE BAYES CLASSIFIER FOR MALWARE DETECTION USING FPGA IMPLEMENTATION

LEE MING YI

UNIVERSITI TEKNOLOGI MALAYSIA

N-GRAM FEATURE EXTRACTION AND NAÏVE BAYES CLASSIFIER FOR MALWARE DETECTION USING FPGA IMPLEMENTATION

LEE MING YI

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

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2022

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

In preparing this thesis, I was in direct contact with many people, academicians, researchers, and classmates. They all have contributed towards my basic understanding and toward the progress that I made. Particularly, I would like to express my sincere appreciation to my thesis supervisor Dr. Ismahani Binti Ismail, for her comments and recommendations on this thesis. I also want to express my gratitude to the School of Electric Engineering in UTM and to all its member’s staff for the guidance that they gave to me, and above all to my dear family and friends for supporting me all the time during my study.

ABSTRACT

Nowadays malicious software, or commonly known as malwares, play a very critical role in almost every network intrusion attack that attempts to harm the connected devices. Thus, installing malware detection systems to protect the network environment has become even more imperative. Naïve Bayes classifier is a probabilistic supervised machine learning algorithm that can be launched on most general-purpose devices to solve a wide range of classification problems, including malware detection. Apart from the classifier, a good feature extractor is important to improve the performance and reliability of the classifier model. However, when it comes to real time applications, the general-purpose devices are limited in terms of their computational throughput. Therefore, the aim of this project is to implement n-gram feature extractor and Naïve Bayes classifier on hardware environments. To improve the throughput and latency of the malware detection, parallel processing capability of field-programmable gate array (FPGA) has been exploited whereby multiple processing units have been designed for the inference module to be implemented on the hardware. Besides, the inference module is designed to be pipelined with six stages. Other than that, hardware-friendly algorithms which have implemented base 2 logarithm transformation and floating-point to fixed-point conversion are used in this study. From the result, both software and hardware designs have obtained similar accuracy of 99.18% on the test dataset. Besides, it is found out that the higher number of parallel processing units, n in this design leads to higher throughput, resource utilization, power consumption, and energy efficiency for malware detection. Hardware design with n = 62 is the optimal design in this project, as it has achieved the highest value of throughput and energy efficiency at the same time.

ABSTRAK

Pada masa ini, perisian hasad memainkan peranan yang sangat penting dalam hampir setiap serangan pencerobohan rangkaian yang cuba merosakkan peranti yang disambungkan. Oleh itu, pemasangan sistem pengesanan perisian hasad untuk melindungi persekitaran rangkaian menjadi semakin penting. Pengelas Naïve Bayes merupakan algoritma pembelajaran mesin diselia kebarangkalian yang boleh dilancarkan pada kebanyakan peranti untuk menyelesaikan pelbagai masalah pengelasan, termasuk pengesanan perisian hasad. Selain pengelas, pengekstrak ciri yang baik adalah penting untuk meningkatkan prestasi dan kebolehpercayaan model pengelas. Walau bagaimanapun, peranti tujuan am adalah terhad dari segi daya pengiraan apabila bincang mengenai aplikasi masa nyata. Oleh itu, projek ini bertujuan untuk melaksanakan pengekstrak ciri n-gram dan pengelas Naïve Bayes pada FPGA. Untuk meningkatkan daya pemprosesan dan kependaman pengesanan perisian hasad, keupayaan pemprosesan selari FPGA akan dieksploitasi dimana berbilang unit pemprosesan akan direka bentuk untuk modul inferens dilaksanakan pada perkakasan. Selain itu, modul inferens yang setara akan direka bentuk dengan seni bina talian paip 6-tahap. Tambahan pula, algoritma mesra perkakasan yang telah melaksanakan transformasi logaritma asas dua dan penukaran titik terapung kepada titik tetap telah digunakan dalam kajian ini. Daripada hasilnya, kedua-dua reka bentuk perisian dan perkakasan telah memperoleh ketepatan yang sama sebanyak 99.18% pada set data ujian. Selain itu, kajian ini menunjukkan bahawa bilangan unit pemprosesan selari, n yang lebih tinggi membawa kepada daya pemprosesan, penggunaan sumber, penggunaan kuasa dan kecekapan tenaga yang lebih tinggi untuk pengesanan perisian hasad. Reka bentuk perkakasan dengan n = 62 adalah reka bentuk optimum dalam projek ini, kerana ia telah mencapai nilai daya pemprosesan dan kecekapan tenaga tertinggi pada masa yang sama.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

APIs	-	Application Programming Interfaces
ASIC	-	Application-Specific Integrated Circuit
BN	-	Bayesian Network
BRAM	-	Block Random Access Memory
CNN	-	Convolutional Neural Network
CPD	-	Critical Path delay
CPU	-	Central Processing Unit
DSP	-	Digital Signal Processing
FF	-	Flip Flop
FN	-	False Negative
FP	-	False Positive
FIFO	-	First-In First-Out
FPGA	-	Field-Programmable Gate Array
IO	-	Input-output
GPU	-	Graphics Processing Unit
IC3	-	Internet Crime Complaint Center
IDS	-	Intrusion Detection Systems
IG	-	Information Gain
J48	-	C4.5 Decision Tree Variant
KNN	-	K-Nearest Neighbors
LMT	-	Logistic Model Trees
LUT	-	Lookup Table
MMCM	-	Mixed-Mode Clock Managers
MP	-	Multilayer Perceptron
NB	-	Naïve Bayes
NBC	-	Naïve Bayes Classifier
NLP	-	Natural Language Processing
PL	-	Programmable Logic
PLL	-	Phase-Locked Loop
PS	-	Processor Subsystem

PU	-	Processing Unit
RAM	-	Random Access Memory
RF	-	Random Forest Tree
ROM	-	Read-Only Memory
SIMD	-	Single Instruction Multiple Data
SLR	-	Simple Logistic Regression
SMO	-	Sequential Minimal Optimization
SoC	-	System on a Chip
SPI	-	Serial Peripheral Interface
SRAM	-	Static Random Access Memory
SVM	-	Support Vector Machine
TN	-	True Negative
TP	-	True Positive

CHAPTER 1

INTRODUCTION

1.1 Problem Background

Malicious software, or more commonly known as malware, is any computer program that can lead to damage on computer hosts. Some classes of malware include all sorts of trojan horses, worms, viruses, rootkits, spyware, and scareware [1]. Most common action from the malware is it will replicate speedily by contaminating any system or any host files inside the computer. Besides, it has the capability to multiply innumerably to generate new obfuscated virus code [2]. Based on the internet crime complaint center (IC3) that was created in 2000 [3], up to 4,883,231 complaints have been received by the center since its creation. From those reported, up to 49,711 reports have been received in its first year of operation. Further details about the numbers of complaints and the amount of money lost by such incidents are presented in [4]. According to the United States council of Economic Advisers, malware has cost the U.S economy between $57 billion and $109 billion USD just in 2016 [5].

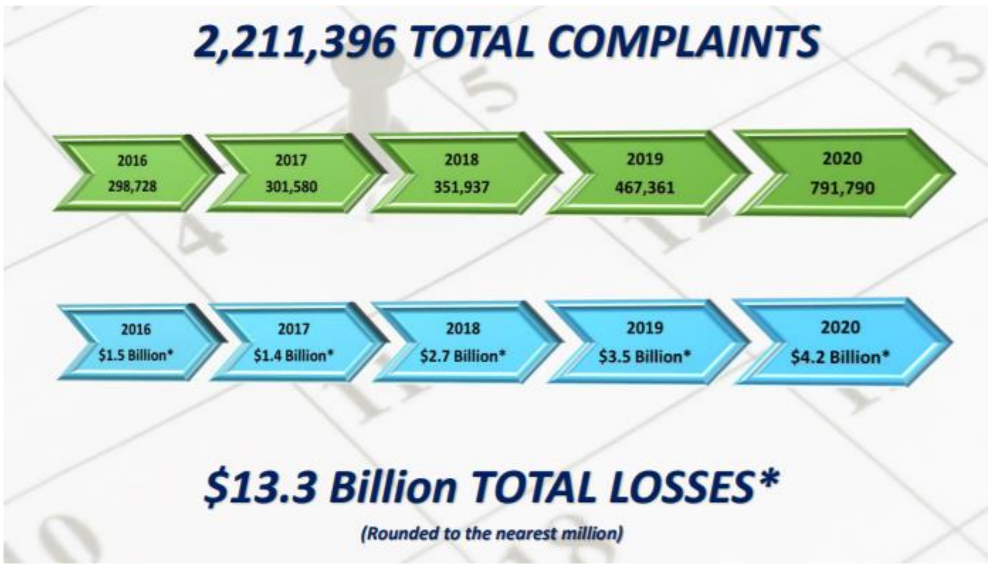

Figure 1.1 Total complaints and amount of money lost over the years between 2016 and 2020 recorded by IC3 [4]

The demand for reliable and accurate malware detection techniques is high to get rid of the relentless attacks. Conventional detection using behavioral, anomaly, and rules-based pattern detections for intrusion detection system (IDS) is unable to provide efficient detection as the malware keeps changing in time. The system needs to frequently increase the malware database so that it can keep up to date with the latest malware samples. However, this will lead to longer scanning time as the database is greater in size, leading to performance drop in terms of throughput [6]. Machine learning classifier such as Naïve Bayes algorithm is a better choice as it is widely used to deal with large volumes of data and capable of producing good results when it comes to Natural Language Processing (NLP) tasks such as sentimental analysis or malware detection. It is a fast classification algorithm and not complicated to understand. Meanwhile, n-gram feature extractor is commonly used together with machine learning classifier, which serves to extract the input features for the NLP task.

This paper is discussing the implementation of machine learning using Naïve Bayes classifier and n-gram feature extractor on FPGA. The hardware design will be improved by exploring the hidden concurrency and mapping it into parallel hardware using pipeline and Single Instruction Multiple Data (SIMD) approach. Then, trade-off analysis on throughput, resources usage, power consumption, and energy efficiency will be carried out with different numbers of parallel processing units on FPGA to obtain the optimum design. Ultimately, an FPGA with malware detection capability will be produced, which allows it to work smoothly in the background to protect the connected devices as a standalone device with better performance compared to CPU.

1.2 Problem Statement

Nowadays, processing the data to ensure safe connection is not the only challenge that we have, as we also aim to improve the user experience by securing the connection with a minimum delay. But as we know, in general computing systems the computing resources were shared among several applications, thus the performance of the algorithm in terms of the overall speed will be affected as the CPU resource is being shared. Due to the CPU limitation, even if the target algorithm is considered as a lite algorithm in terms of its computing time, it is still not efficient enough to handle the real time processing of data especially when the dataset is big enough. Besides, CPUs are optimized for sequential processing, which means they lack sufficient support for parallelism, and this limits their high-speed processing capabilities. Thus, FPGA is a better choice as it inherits parallelism which allows the algorithm to be inferenced in parallel and improve the throughput performance.

Although FPGA can exploit its parallel processing capability to enhance the inference performance and improve the throughput, it will also lead to higher resource utilization and power consumption at the same time. Therefore, tradeoff analysis is required so that optimum design can be selected to fit the purpose. This means that blindly pursuing FPGA design with more parallel processing units might not be a wise choice as resource utilization and power consumption are also the key considerations when designing the hardware. Besides, not all parallelization can result in speed-up. As there are more threads, parallel slowdown can occur as communication time on each other might be increased and longer time is needed to wait for resource access [7]. Therefore, design space exploration is needed so that more insight can be explored regarding the number of parallel processing units and its impact towards the maximum frequency, latency, resource utilization, and power consumption.

1.3 Research Objectives

The objectives of the research are:

(a) To implement n-gram feature extractor and Naïve Bayes classifier as FPGA hardware design for malware detection.

(b) To validate the correctness functionality of the implemented hardware design with the software version in terms of accuracy.

(c) To compare the performance of proposed design in terms of throughput, resources utilization, power consumption, and energy efficiency when different numbers of parallel processing units are used on FPGA.

1.4 Scope

For this project, the scopes covered are as follows:

- Proving whether Naïve Bayes algorithm is efficient in identifying malicious software or not is beyond the scope.
- The dataset that will be used in training and testing of malware detection is being processed from the packet payload.
- Supervised machine learning will be used for two-class malware detection.
- Implementation of n-gram feature extractor on FPGA is for the inference part only whereas the Naïve Bayes algorithm will be implemented for both training and inference part.
- The version of Vivado tool involved in this experiment for design, synthesize, and simulate the project is 2019.1, whereas Xilinx Zynq-7000 FPGA board is used as the target device in Vivado.
- The computer device that carries out the whole experiment including the benchmarking business follows these specifications: Intel(R) Core (TM) i7-1185G7, 3.00 GHz, 16 GB RAM, 7.8 GB Intel(R) Iris(R) Xe Graphics, with 64-bit version of Windows 10.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews about the malware background, Naïve Bayes algorithm, n-gram, and FPGA, as well as the recent similar works that were addressed by other researchers before. Chapter 3 explains the methodology used in this project to implement the design. Chapter 4 describes the results and discussion. Chapter 5 summarizes the proposed works.

REFERENCES

[1] E. G. Dada, J. S. Bassi, Y. J. Hurcha and A. H. Alkali, "Performance Evaluation of Machine Learning Algorithms for Detection and Prevention of Malware Attacks," *IOSR Journal of Computer Engineering,* vol. 21, no. 3, pp. 18-27, 2019.

[2] P. Szor, The Art of Computer Virus Research and Defense, Addison-Wesley, 2005.

[3] "The FBI's Internet Crime Complaint Center (IC3) Marks Its 20th Year," FBI National Press Office, 8 May 2020. [Online]. Available: https://www.fbi.gov/news/pressrel/press-releases/the-fbis-internet-crime-complaint-center-ic3-marks-its-20th-year. [Accessed 31 Jan 2022].

[4] "Internet Crime Report 2020," 2021. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf. [Accessed 31 Jan 2022].

[5] K. Arunlal, "Impact of Malware in Modern Society," *International Journal of Scientific Research and Engineering Development,* vol. 2, no. 3, pp. 593-600, 2019.

[6] X. Zhang, A. Ramachandran, C. Zhuge, D. He, W. Zuo, Z. Cheng, K. Rupnow and D. Chen, "Machine learning on FPGAs to face the IoT revolution," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* 2017.

[7] K. Pugdeethosapol, Z. Jin, D. Rider and Q. Qiu, "Accelerating Block-Circulant Matrix-Based Neural Network Layer on a General Purpose Computing Platform: A Design Guideline," in *FICC 2020,* San Francisco, 2020.

[8] M. Christodorescu and S. Jha, "Static analysis of executables to detect malicious patterns," in *SSYM'03: Proceedings of the 12th conference on USENIX Security Symposium,* 2003.

[9] N. Miramirkhani, M. P. Appini, N. Nikiforakis and M. Polychronakis, "Spotless Sandboxes: Evading Malware Analysis Systems Using Wear-and-Tear Artifacts," in *2017 IEEE Symposium on Security and Privacy (SP),* 2017.

[10] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection," *Hum.-Centric Comput. Inf. Sci.,* vol. 8, no. 1, p. 3, 2018.

[11] O. A. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access,* vol. 8, pp. 6249-6271, 2020.

[12] E. Gandotra, D. Bansal and S. Sofat, "Malware Analysis and Classification: A Survey," *Journal of Information Security,* vol. 5, no. 2, 2014.

[13] B. Cakir and E. Dogdu, "Malware classification using deep learning methods," in *ACMSE '18: Proceedings of the ACMSE 2018 Conference*, 2018.

[14] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2017.

[15] A. C. Müller and S. Guido, Introduction to Machine Learning with Python, O'Reilly Media, Inc., 2016.

[16] A. Z. Mohd. Zuki, "FPGA implementation of naive bayes classifier for network security," *UTM Institutional Repository,* 2018.

[17] I. H. Witten, E. Frank and M. A. Hall, Data Mining Practical Machine Learning Tools and Techniques 3rd Edition, Elsevier Inc., 2011.

[18] C. M. Bishop, Pattern Recognition and Machine Learning, Springer New York, 2006.

[19] C. Eagle, The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler, No Starch Press, 2008.

[20] F. S. M. Alkhafaji, W. Z. W. Hasan, M. M. Isa and N. Sulaiman, "Robotic Controller: ASIC versus FPGA—A Review," *Journal of Computational and Theoretical Nanoscience,* vol. 15, pp. 1-25, 2018.

[21] F. B. Muslin, L. Ma, M. Roozmeh and L. Lavagno, "Efficient FPGA Implementation of OpenCL High-Performance Computing Applications via High-Level Synthesis," *IEEE Access,* vol. 5, pp. pp. 2747-2762, 2017.

[22] J. Robinson, "FPGAs, Deep Learning, Software Defined Networks and the Cloud: A Love Story Part 1," Medium, 12 Nov 2017. [Online]. Available: https://jamal-robinson.medium.com/fpgas-deep-learning-software-defined-

networks-and-the-cloud-a-love-story-part-1-c685dc6b657b. [Accessed 31 Jan 2022].

[23] National Instruments Corp., "Understanding Parallel Hardware: Multiprocessors, Hyperthreading, Dual-Core, Multicore and FPGAs," 11 Dec 2011. [Online]. Available: http://www.ni.com/tutorial/6097/en/. [Accessed 31 Jan 2022].

[24] "Zynq-7000 SoC Data Sheet: Overview(DS190)," Xilinx, Inc, 2 July 2018. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ds190-Zynq-7000-Overview. [Accessed 31 Jan 2022].

[25] L. A. Tambara, F. L. Kastensmidt, N. H. Medina, N. Added, V. A. P. Aguiar, F. Aguirre, E. L. A. Macchione and M. A. G. Silveira, "Heavy Ions Induced Single Event Upsets Testing of the 28 nm Xilinx Zynq-7000 All Programmable SoC," in *2015 IEEE Radiation Effects Data Workshop (REDW)*, Boston, 2015.

[26] Z. Xue, J. Wei and W. Guo, "A Real-Time Naive Bayes Classifier," *IEEE Access,* vol. 8, pp. 40755-40766, 2020.

[27] Y. K. Al Hussein, "Hardware Implementation of Naive Bayes Classifier for Malware Detection," *UTM Institutional Repository,* 2021.

[28] F. Zhang and T. Zhao, "Malware Detection and Classification Based on N-Grams Attribute Similarity," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 2017.

[29] Q. Wang, Y. Li, B. Shao, S. Dey and P. Li, "Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA," *Neurocomputing*, vol. 221, pp. 146-158, 2017.

[30] "Capture files from Mid-Atlantic CCDC," NETRESEC, [Online]. Available: https://www.netresec.com/?page=MACCDC. [Accessed 31 Nov 2021].

[31] "2017-08-01 - Rig EK from the HookAds Campaign Sends Dreambot," MALWARE-TRAFFIC-ANALYSIS.NET, [Online]. Available: https://www.malware-traffic-analysis.net/2017/08/01/index.html. [Accessed 31 Nov 2021].

[32] "2018-08-07 - HookAds Rig EK Pushes Azorult, Azorul Pushes Smokeloader," MALWARE-TRAFFIC-ANALYSIS.NET, [Online]. Available:

https://www.malware-traffic-analysis.net/2018/08/07/index.html. [Accessed 31 Nov 2021].

[33] "The Jupyter Notebook," Jupyter, 2022. [Online]. Available: https://jupyter.org/. [Accessed 31 Jan 2022].