FPGA IMPLEMENTATION OF NAIVE BAYES CLASSIFIER FOR NETWORK SECURITY

AHMAD ZULZHA FRI BIN MOHAMAD ZUKI

UNIVERSITI TEKNOLOGI MALAYSIA
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

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>ii</td>
<td></td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iii</td>
<td></td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iv</td>
<td></td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>ABSTRAK</td>
<td>vi</td>
<td></td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>vii</td>
<td></td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
<td></td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>xi</td>
<td></td>
</tr>
</tbody>
</table>

1 INTRODUCTION 1

1.1 Problem Background 1

1.2 Problem Statement 4

1.3 Objective 4

1.4 Scope 5

1.5 Organization 5

2 LITERATURE REVIEW 6

2.1 Introduction 6

2.2 Malware Types 6

2.3 Detection Methods 7

2.4 Supervised and Unsupervised Learning 10

2.5 Need of Machine Learning for Malware Detection 12

2.6 Related Works 12

2.7 Naive Bayes 14

2.8 N-Gram Analysis 16

2.9 FPGA Platform for Malware Detection Processing 18

2.10 Related Works on NB Architecture on FPGA 19

2.11 Limitations and Research Gaps 20
3 RESEARCH METHODOLOGY

3.1 Introduction
3.2 System Overview of Proposed Work
3.3 Experimental setup
   3.3.1 RTL Architecture
   3.3.2 Tools and Platforms
   3.3.3 Dataset
   3.3.4 Performance Evaluation
3.4 Project Management
   3.4.1 Research Planning and Schedule
3.5 Chapter Summary

4 RESULTS AND DISCUSSION

4.1 Design Implementation
   4.1.1 NB Malware Classifier using WEKA
   4.1.2 Naïve Bayes Classifier FPGA Implementation
4.2 The Results of proposed work
4.3 Chapter Summary

5 CONCLUSION

5.1 Project Accomplishment
5.2 Future Works

REFERENCES
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>N–grams for different values of n = 1, 2, 3, 4</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Features n–gram after classwise document frequency</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Values for feature after feature reduction stage</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Training and testing set used</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Evaluation parameters for different n-grams average values</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Evaluation parameters for different classifiers mean values</td>
<td>36</td>
</tr>
<tr>
<td>4.3</td>
<td>Naïve Bayes classifier evaluation metrics</td>
<td>36</td>
</tr>
<tr>
<td>4.4</td>
<td>Resource Utilization in NB malware detection classifier</td>
<td>39</td>
</tr>
<tr>
<td>4.5</td>
<td>Logic utilization in NB malware detection classifier</td>
<td>40</td>
</tr>
<tr>
<td>4.6</td>
<td>Prediction results</td>
<td>40</td>
</tr>
<tr>
<td>4.7</td>
<td>Hardware performance result</td>
<td>41</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Conventional malware detection architecture with feature reduction</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Virtex-7 VC709 board</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>Connections between FPGA and Ethernet</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>System Overview</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Flow of the hardware accelerator</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>4-gram generation from known signature</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Learning process incorporated by known malware signatures</td>
<td>24</td>
</tr>
<tr>
<td>3.5</td>
<td>Training dataset based on selected features [63]</td>
<td>24</td>
</tr>
<tr>
<td>3.6</td>
<td>Flowchart for NB Model implementation</td>
<td>28</td>
</tr>
<tr>
<td>3.7</td>
<td>Dataset preparation sampling traffic</td>
<td>30</td>
</tr>
<tr>
<td>3.8</td>
<td>Gantt chart FYP1</td>
<td>33</td>
</tr>
<tr>
<td>3.9</td>
<td>Gantt chart FYP2</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>Architecture of Naive Bayes classifier</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Architecture of LUT and accumulator</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Architecture of NB_Decision module system</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>Snippet of simulation result of the system</td>
<td>41</td>
</tr>
</tbody>
</table>
# LIST OF ABBREVIATIONS

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


44(2): 6:1–6:42.


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


